Journal of Cheminformatics

REVIEW

Open Access

Towards reproducible computational drug discovery

Nalini Schaduangrat[1†], Samuel Lampa[2†], Saw Simeon[3†], Matthew Paul Gleeson[4*], Ola Spjuth[2*] and Chanin Nantasenamat[1*]

Abstract

The reproducibility of experiments has been a long standing impediment for further scientific progress. Computational methods have been instrumental in drug discovery efforts owing to its multifaceted utilization for data collection, pre-processing, analysis and inference. This article provides an in-depth coverage on the reproducibility of computational drug discovery. This review explores the following topics: (1) the current state-of-the-art on reproducible research, (2) research documentation (e.g. electronic laboratory notebook, Jupyter notebook, etc.), (3) science of reproducible research (i.e. comparison and contrast with related concepts as replicability, reusability and reliability), (4) model development in computational drug discovery, (5) computational issues on model development and deployment, (6) use case scenarios for streamlining the computational drug discovery protocol. In computational disciplines, it has become common practice to share data and programming codes used for numerical calculations as to not only facilitate reproducibility, but also to foster collaborations (i.e. to drive the project further by introducing new ideas, growing the data, augmenting the code, etc.). It is therefore inevitable that the field of computational drug design would adopt an open approach towards the collection, curation and sharing of data/code.

Keywords: Reproducibility, Reproducible research, Drug discovery, Drug design, Open science, Open data, Data sharing, Data science, Bioinformatics, Cheminformatics

Introduction

Traditional drug discovery and development is well known to be time consuming and cost-intensive encompassing an average of 10 to 15 years until it is ready to reach the market with an estimated cost of 58.8 billion USD as of 2015 [1]. These numbers are a dramatic 10% increase from previous years for both biotechnology and pharmaceutical companies. Of the library of 10,000 screened chemical compounds, only 250 or so will move on to further clinical testings. In addition, those that are tested in humans typically do not exceed more than 10 compounds [2]. Furthermore, from a study conducted during 1995 to 2007 by the Tufts Center for the Study of Drug Development revealed that out of all the drugs that make it to Phase I of clinical trials, only 11.83% were eventually approved for market [3]. In addition, during 2006 to 2015, the success rate of those drugs undergoing clinical trials was only 9.6% [4]. The exacerbated cost and high failure rate of this traditional path of drug discovery and development has prompted the need for the use of computer-aided drug discovery (CADD) which encompasses ligand-based, structure-based and systems-based drug design (Fig. 1). Moreover, the major side effects of drugs resulting in severe toxicity evokes the screening of

*Correspondence: paul.gl@kmitl.ac.th; ola.spjuth@farmbio.uu.se; chanin.nan@mahidol.edu
†Nalini Schaduangrat, Samuel Lampa and Saw Simeon contributed equally to this work
[1] Center of Data Mining and Biomedical Informatics, Faculty of Medical Technology, Mahidol University, 10700 Bangkok, Thailand
[2] Department of Pharmaceutical Biosciences, Uppsala University, 751 24 Uppsala, Sweden
[4] Department of Biomedical Engineering, Faculty of Engineering, King Mongkut's Institute of Technology Ladkrabang, 10520 Bangkok, Thailand
Full list of author information is available at the end of the article

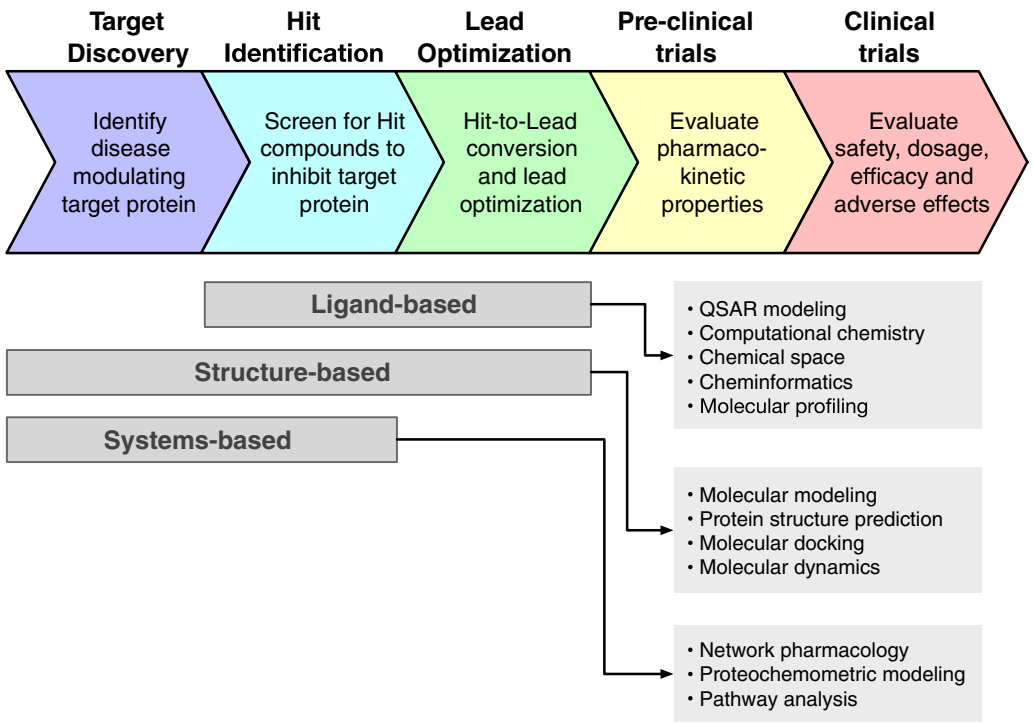

Fig. 1 Schematic summary of the drug discovery process overlayed with corresponding computational approaches

ADMET (adsorption, distribution, metabolism, excretion and toxicity) properties at the early stage of drug development in order to increase the success rate as well as reduce time in screening candidates [5]. The process of CADD begins with the identification of target or hit compound using wet-lab experiments and subsequently via high-throughput screening (HTS). In particular, the typical role of CADD is to screen a library of compounds against the target of interest thereby narrowing the candidates to a few smaller clusters [6]. However, owing to the high requirement of resources for CADD coupled with its extensive costs, opens the door for virtual screening methods such as molecular docking where the known target of interest is screened against a virtual library of compounds. Although this method is highly effective, a crystal structure of the target of interest remains the main criteria required of this approach in generating an in silico binding model. However, in the absence of a crystal structure, homology modeling or de novo prediction models can still be obtained against the large library of compounds to acquire compounds with good binding affinity to the target [7] which are identified as hits and could be further developed as lead compounds [8]. A conceptual map on the experimental and computational methodologies as applied to the drug discovery process is summarized in Fig. 2.

In recent years, the expansion of data repositories including those with chemical and pharmacological data sets, has significantly increased the availability of large-scale open data for drug discovery. In addition, more data are being deposited into these domains on a daily basis, with some repositories containing tens of millions of compounds (e.g. PubChem and ZINC databases) [9]. The availability of such large-scale data sets has had a significant impact on the drug discovery process. Moreover, this process may help address many of the unmet needs in drug discovery and design such that the access to these data may help with the rapid identification of compounds to validate targets or profile diseases which will further encourage the development of new tools and predictive algorithms. Furthermore, large bioactivity data sets can be used for the identification of quantitative structure–activity relationships (QSAR) or classification models, allowing prediction of compound activities from their structures. Such predictions can contribute to molecular target elucidation, drug ADMET prediction and potential drug repurposing [10]. However, with all the predictive methods, the quality and relevance of the data acquired are paramount in determining the accuracy and applicability of the resulting models. Therefore, as data sets become more readily available due to the open science initiative, the emphasis has now moved towards quality, rather than the quantity of raw data. Indeed, many

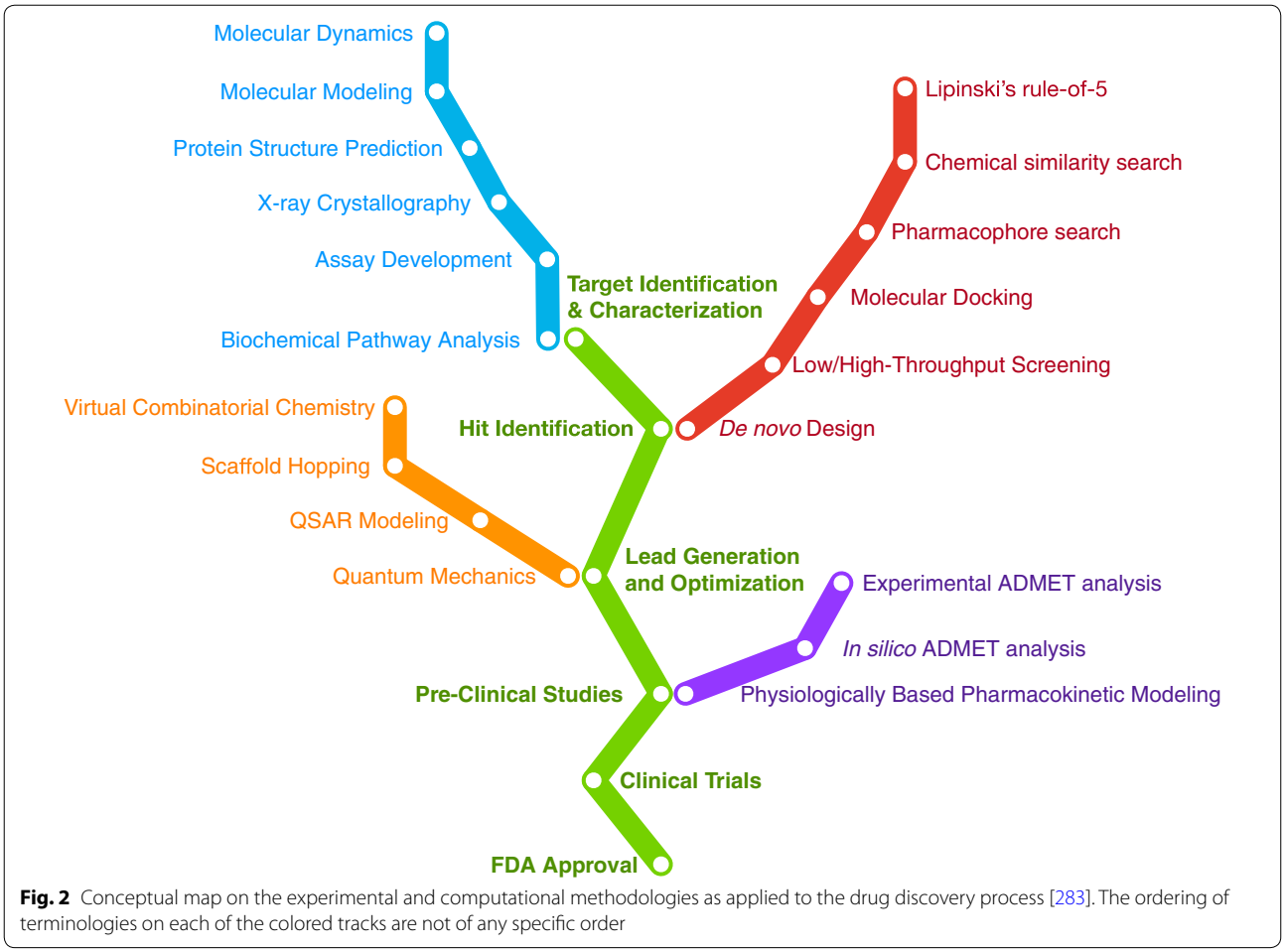

Fig. 2 Conceptual map on the experimental and computational methodologies as applied to the drug discovery process [283]. The ordering of terminologies on each of the colored tracks are not of any specific order

analyses have been published assessing the quality of screening libraries that identify compounds responsible for many of the false-positive results [11, 12] or investigate compound structure accuracy in various repositories [13, 14]. Hence, any progress made within just this one area will have a profound impact on improving the development of novel and safe drugs. Nevertheless, with the increasingly rapid growth of these public data sources therefore efforts in ensuring the quality and interoperability will be essential for maximizing the utilization of data.

In the midst of big data expansion (i.e. borne from omics data) that are available for computational drug discovery, proper efforts for ensuring the quality of these data are made possible through data curation and preprocessing as carried out by database and repository providers. Workflows and pipelines in the form of markup languages, codes or software tools have become instrumental in ensuring the reproducibility of computational research as it helps to materialize the actual steps and procedures taken during the entire computational study. Discussion on the availability and current efforts undertaken in the field of computational drug discovery (i.e. also encompassing bioinformatics and cheminformatics) in regards to research reproducibility is provided in this review article. During the revision phase of this manuscript submission, an excellent commentary article by Clark [15] addressing the importance of reproducibility in cheminformatics was recently published. Moreover, a blog post by cheminformatic researchers [16] also reaffirmed the significance of this point and the timely manner of the topic of this review article so as to encourage further developments and paradigm shifts in computational drug discovery and neighboring fields (e.g. bioinformatics and cheminformatics) pertaining to research reproducibility.

Research documentation

Scientific experiments have long preceded digital logging of laboratory activities. Documentation of experimental results has traditionally been kept within the confinement of paper-based notebooks whereby the scientific

benefits of which is to allow subsequent reproduction of the documented experiment, while its legal use is to serve as a proof of inventorship [17]. The reporting of science is fundamental to the scientific process, which, if done clearly and accurately, can help advance knowledge and its reproducibility [18]. All professionals working in life sciences are familiar with the importance of keeping laboratory notebooks. Although, science as a field has advanced over the centuries, the methods of recording data (i.e. in a paper-based, inked and bound notebook) has remained unchanged. In addition, the current reproducibility crisis has put the spotlight on data recording. Therefore, unsurprisingly, many industries and laboratories are now shifting to a digital form of record keeping, the electronic laboratory notebooks (eLNs) [19].

eLNs have been introduced as a digital alternative to the paper-based version but with enhanced capabilities such as search capability, integration with instrumentation, etc. [20]. Scientists are increasingly adopting the use of eLNs in their research laboratories owing to the inherent need to organize the growing volume of biological data [21]. Recently, Schnell [22] had proposed ten simple rules for a computational biologist's laboratory notebook, which highlights the importance of documenting all the minute details that were carried during the course of project from start to finish (i.e. applicable to all scientific disciplines) while also making use of version control, virtual environments and containers (i.e. applicable to computational disciplines). Particularly, which software version was used, which parameter values were used, which specific algorithms and specific options were utilized for the calculation, etc. Moreover, scientists are making these notebooks publicly available so as to support the open science initiative (i.e. also termed "open notebook science") [23, 24] and in doing so foster the sharing of unpublished experimental data and analysis (i.e. known as "dark data"). These interactive notebooks (i.e. also known as iPython/Jupyter notebooks) have evolved to the point that it is possible for the code used to perform the data analysis to be shown alongside the explanatory text and visualizations (e.g. images, plots, etc.), thereby affording easy comprehension of the experimental results and its underlying code, thus facilitating reproducible research.

The iPython notebook was created in 2001 by Fernando Perez and has since evolved to the more general and powerful Jupyter notebook [25] with support for more than 40 programming languages (e.g. Python, R, Javascript, Latex, etc.). For the sake of data sharing, it is common practice to store the Jupyter notebooks (i.e. used hereon to also refer to the iPython notebook) on GitHub (i.e. or other web repository such as BitBucket). Such notebook files can then be rendered as static HTML via the nbviewer [26]. Recently, GitHub also made it possible for Jupyter notebook files to render directly on its repositories. Owing to the static nature of the rendered notebook the resulting HTML is consequently not interactive and therefore not amenable to modifications. A first step towards solving this limitation is made by the Freeman lab at Janelia Research Campus in their development of binder [27], a web service that converts Jupyter notebook files hosted on GitHub to executable and interactive notebooks. Google CoLaboratory [28] is another interface which utilizes the Jupyter notebook environment for the dissemination of research and education. Google Colaboratory is a free platform whereby projects can be run completely on the cloud, without the need for any software setups while the *"notes"* are stored entirely on Google Drive and can be easily accessed and shared.

At the other end of the spectrum are cloud-based word processors such as Google Docs, Overleaf, ShareLatex and Authorea that facilitate collaborative writing of experimental findings and results in the form of manuscripts, books and reports. A distinctive feature of these applications is the possibility for several users (i.e. who can be physically located in different parts of the world) to be able to work on the same document at the same time. Most of these web applications serve as only word processors that house the text of a manuscript but does not allow integration with the Jupyter notebook. In fact, only Authorea integrates interactive Jupyter notebooks (i.e. also hosted by Authorea) into their application so that users can play around with the parameters and come up with customized figures and plots.

Science of reproducible research

Reproducibility crisis

According to an online survey conducted by Nature of 1576 researchers, it was revealed that 52% of researchers agreed that there is a significant reproducibility crisis while 38% agreed that there is a slight crisis. On the other hand, 3% of those surveyed do not think that there is such a reproducibility crisis while 7% of researchers are not aware of its very existence [29]. These results suggests confusing viewpoints as to what constitutes reproducible research. In addition, when asked to identify the problem associated with this crisis, the same survey reported over 60% of respondents believe that the pressure to publish and selective reporting contributed to the problem. Furthermore, lesser contributing factors reported were unable to replicate the work in the lab, low statistical power and obstacles such as reagent variability or the use of specific techniques that are difficult to replicate.

The concept of reproducibility in science depends on the dissemination of knowledge and the reproducibility of results. To facilitate this, the accurate and clear

reporting of science should be a fundamental part of the scientific process. Plavén-Sigray et al. [18] believe that the readability of a scientific research is one of the main factors for reproducible and accessible literature. From a compilation of 709,577 abstracts from 123 scientific journals published between 1881 and 2015 on biomedical and life sciences coupled with readability formulas, the authors concluded that the readability of scientific literature has been decreasing over time. Lower readability could in turn discourage accessibility, particularly from non-specialists and the importance of comprehensive texts in regards to the reproducibility crisis cannot be ignored.

Another aspect of the reproducibility crisis can be seen during the data analysis whereby it can be difficult for researchers to recognize *p*-hacking also known as data dredging [30] (i.e. the phenomenon where researchers select statistical analysis which portray insignificant data as significant) due to confirmation and hindsight biases which encourage the acceptance of preconceived outcomes that fit expectations [31]. Hence, there is an increased concern that most published articles are based on false or biased results [32]. In addition, several studies have pointed out that the high rate of non-replicable discoveries is a consequence of basing conclusive findings on a single study assessed via only the statistical significance (i.e. the *p*-value) [32–34]. Therefore, in order to combat this disturbing trend, striving towards the FAIR (Findable, Accessible, Interoperable and Reproducible) [35] principle in research practices can help to ensure that models and studies are FAIR for them to be consumed and integrated on-demand. Hence, studies using open data derived from analysis according to the FAIR principles, will pave the way towards iteratively better science with higher confidence in the reproducibility of research [36].

Reproducibility versus replicability

It is important to note that the terminology found across the scientific literature such as reproducibility, replicability, reusability, recomputibility and their associated definitions are not standardized and thus has led to confusion regarding their usage. *"Reproducibility"* has been defined in the dictionary as *"the ability to produce, form or bring about again, when repeated"* [37]. In the context of computational research, the term *"reproducible research"* was first coined by Jon Claerbout in 1990, the geophysicist who implemented the standard for maintaining and building executable programs from the source code leading to the construction of computational results known as the Stanford Exploration Project in published articles [38]. An important issue for reviewers and authors alike, reproducibility acts as a bedrock principle for the validation in experimental scientific research. However, with such emphasis placed on reproducibility in experimental sciences, two conspicuous discrepancies were highlighted by Casadevall and Fang [39]. First, while the work conducted and published by scientists are expected to be reproducible, most scientists do not partake in replicating published experiments or even read about them. Furthermore, despite the obvious prerequisite in most reputable journals whereby, all methods must be reported in adequate detail so as to allow replication, no manuscripts highlighting replicated findings without the discovery of something novel are published. Thus, the reproducibility of any given published research is assumed, yet only rarely is that notion tested. In actuality, the reproducibility of experiments are only highlighted when a given work is called into question [40]. Hence, the consistency of this basic supposition relies heavily on integrity of the authors publishing the results and the trust afforded to them by the publishers and readers [39]. Ironically, suspicions of data falsification are sometimes heightened when results are deemed as *"too good to be true"* [40]. Therefore, this replication debate provides an opportunity to redefine the differences between replicability and reproducibility.

As such, strict definitions of both terms are also available and could be useful in discerning slight differences that occur by either repeating or reproducing an experiment/workflow. According to the *Guide to the expression of uncertainty in measurement* [41], reproducibility is defined as the *"closeness of the agreement between the results of measurements of the same measure and carried out under changed conditions of measurement"* while repeatability or replicability is defined as the *"closeness of the agreement between the results of successive measurements of the same measure and carried out under the same conditions of measurement"*. Although the mismatch of both terms is not so critical in some cases, it is important to clarify the main differences. For example, if the experiment/model conditions are close or identical, they should be successfully repeated (i.e. repeatability or replicability). On the other hand, if the experimental/model conditions are changed to some degree, the exact or close match results may not be obtained but the methodology should be sound (i.e. reproducibility).

Reusability versus reliability

In life sciences, the reliability of a published protocol is a pressing matter upon implementation. Reusability is more prevalent in computer science in which codes created by an individual or groups of individuals that are shared on public repositories, can be reused by others as well as facilitate future work to be built upon it. Hence, enabling reusability represents an important catalyst that

would help to advance the field. Conventionally, scientific research relies on results from independent verification. Specifically, when more people verify an observation or hypothesis, the more trustworthy it becomes. A conjecture, on the other hand, without verification is therefore not considered to be well-thought-out. Thus, replication represents an important facet of verification within which theories are confirmed by equating predictions in relation to reality. For computational research however, no established verification practices exist as of yet [42]. Although a research may be reproducible, the quality, accuracy or validity of the published results are not guaranteed. Therefore, simply bringing the notion of reproducibility to the forefront and making it as routine as keeping a laboratory notebook, would help set the stage for a reproducible atmosphere. Encouragingly, the minimum information checklist brought together under the umbrella of the Minimum Information for Biological and Biomedical Investigations (MIBBI) project [43] has helped to ensure that all pertinent data is provided by researchers. Furthermore, bioinformatics software typically involve a wide variety of data formats which can make the execution of replicability a little more difficult. However, softwares pertaining to data exchange and analysis such as the Proteomics Standard Initiative for molecular interactions (PSI-MI) for proteomics [44] and the Biological Pathway Exchange (BioPAX) language [45] representing metabolic and signaling pathways, molecular and genetic interactions and gene regulation networks, have been developed to improve this. In addition, the Workflow4Ever project [46] caters to the same aim using a different approach.

The underlying aim of reproducing any given research/experiment is so that the work being proposed can be extended rather than just to confirm it. It also then, makes perfect sense that the extensibility of methods in the computational realm is taken into account during the design phase [47]. Conducting research can, in this day and age, no longer be a lone endeavour; rather, collaborations have permanently made their way into the sciences. In that respect, many bioinformatic tools have been developed under a joint effort where one group extended the work of another group such as the Bioconductor [48] and Galaxy [49–51] projects. In addition, a tool specifically made for analyzing phylogenetic data, Beast 2 [52] and Beast 2.5 [53], emphasizes modular programming techniques into its software in order to allow the software to be extensible by users. Furthermore, the Jupyter Notebook [25] offers a dynamically updating, error-correcting tool for the publication of scientific work, thus facilitating extensibility. In addition, protocols.io [54] is an open access repository for scientific protocols that allow lab members to write and edit collaboratively.

This debate further behooved the question as to who would benefit from the detailed accumulation of methods in scientific papers or codes shared on various virtual platforms. Perhaps, it would be most advantageous for the new scientist as they can learn to use novel software/protocol without going into too much detail and without having to write the code themselves. In addition, it allows the general public to make use of, and maneuver a minimal working environment while saving time which could possibly provide a fresh perspective to existing research data.

Open Science

In the last decade or so, the sharing of scientific data has been promoted by a growing number of government and funding agencies [55, 56]. As such, open access to data from research networks, governments, and other publicly funded agencies has also been on the rise given the policies that promote them [57]. However, the sharing of data in terms of policies varies dramatically by field of research, country, and agency, yet many of their goals are conjoint. Upon analysis of these policies, Borgman [58] found that the data sharing policies are based on four main features (i.e. reproducible research, making data available to the public, influencing investments in research, and advancing research and innovation). Epistemically, the impulse for the production of new knowledge with the reuse of data through open sources, is the key take away from these arguments [35, 59]. The proposed benefits of sharing can only be accomplished if and when the data is shared and/or reused by others [58]. Hence, *"data sharing"* refers to the idea and implementation of data release and in its simplest form, is the act of making data readily and easily available and accessible [60]. Data sharing thus, encompasses many means of releasing data, while saying little about the usability of those data. Some ways whereby researchers share their data are private exchanges, posting data sets on websites (e.g. GitHub or Figshare); depositing data sets in archives or repositories (e.g. PubChem or ChEMBL); and supplementary materials provided in research articles [61]. Data papers represent a newer avenue in the research field whereby descriptions similar to the *"Methods"* section of a traditional research article are published with greater details regarding the processes used for data collection, experimentation and verification [62, 63].

Furthermore, reproducibility can be seen to critically affect various aspects of research, especially in the field of science [29]. However, these days bioinformatics plays a distinct role in many biological and medical studies [64]. Thus, a great effort must be made to make computational research reproducible. As such, many reproducibility issues that arise in bioinformatics may be due to various

reasons such as version of bioinformatics software, complexity of its pipeline and workflow, technical barriers ranging from insufficient data to hardware incompatibility, etc. [65]. This crisis has been described by Kim et al. [66] whereby the authors compare the hidden reproducibility issues to an iceberg which is only noticed at a fraction of its actual size, highlighting the significant gap between the apparent executable work (i.e. portion of iceberg that can be seen above water) and the necessary effort required to practice (i.e. the full iceberg).

To deal with this reproducibility crisis, Sandve et al. [67] proposed ten simple rules for reproducible computational research, through which the authors encourage researchers to responsibly and consciously make small changes during their computational workflow in order to achieve reproducibility habits that benefit not only the researchers but their peers and the scientific community on the whole. In our humble opinion, one of the most important point from the article stressed the importance of publicly sharing the data and source code so as to foster reproducibility of the work and in turn move science forward. One of the projects that implemented most rules laid out by Sandve et al. is the Bioconductor project [48] which is an open software that encourages collaborations in the fields of computational biology and bioinformatics. In addition, BaseSpace [68] and Galaxy [51] represent examples of both commercial and open-source solutions, that partially fulfill the ten simple rules laid out in the aforementioned review. However, workflow customizations on such environments are not implementable, for example, BaseSpace have strict application submission rules and being cloud based, have to cope with ethical and legal issues [69].

The applications and pipelines in bioinformatics require a substantial effort to configure, therefore container-based platforms, such as Docker [70], have emerged to allow the deployment of individual applications that have an isolated environment for the installation and execution of a specific software, without affecting other parts of the system. In this regard, many docker-based platforms have been produced such as BioContainer [71], a community-driven, open-source project based on the Docker container that can be easily accessed via GitHub; Bio-Docklets [72], a bioinformatics pipeline for next generation sequencing (NGS) data analysis; and Dugong [73], a Ubuntu-based docker that automates the installation of bioinformatics tools along with their libraries and dependencies on alternate computational environments. The above-mentioned platforms utilize the Jupyter Notebook as an integration platform for delivery and exchange of consistent and reproducible protocols and results across laboratories, assisting in the development of open-science. In addition, the Reproducible Bioinformatics Project [74] is a platform that distributes docker-based applications under the framework of reproducibility as proposed by Sandve et al. Furthermore, the more recently established Human Cell Atlas [75] is an ambitious project encompassing more than 130 biologists, computational scientists, technologists and clinicians. Their aim is to help researchers answer questions pertaining to the human body in diverse biological fields. However, to provide maximum impact and continued collaborations, the project will be a part of open science on multiple levels to ensure that the results are of high quality and are technically reproducible. The initiative currently includes members from 5 continents and more than 18 countries, including Japan, Israel, South Africa, China, India, Singapore, Canada and Australia. The work conducted by this initiative in a large-scale international, collaborative and open effort may bring different expertise to the problems and could dramatically revolutionize the way we see our cells, tissues and organs.

Computational reproducibility ecosystem

So the question is, how does one go about making their own research reproducible? For a computational life scientist there are a plethora of resources that are enabling factors for data-driven research and it is the intent of this section to attempt to provide a broad if not extensive coverage. Conceptually, a reproducible ecosystem could be thought of as environments or enabling factors that, on one end, allow the practitioner to archive and share their data and codes while on the other end, allow third-party users to gain access to these resources so that they can build upon them in their own independent projects. A more in-depth coverage of this topic is provided elsewhere [76, 77]. Traditionally, data (and rarely codes) accompanying a research article containing computational component(s) are provided in the Supplementary. In recent years, several web-based services are available as enabling proponents for computational reproducibility as will be discussed hereafter.

Data repository is a relatively general term used to reference a storage site specifically delegated for data depositions. As part of the reproducibility era, many appropriate public data depositories are now available such that, authors are able to deposit their raw research data into discipline specific and community-recognized repositories (i.e. GenBank, PDB, PubChem, etc. for discipline specific and figshare, Dryad digital repository, Zenodo, Open science framework, etc. for general repositories) [78]. The Dryad digital repository [79] is an open, curated, reusable and easily citable resource for scientific data. In addition, Dryad was initiated as a non-profit organization employing the joint data archiving policy (JDAP) for journal submissions with integrations.

Moreover, DryadLab [80], a project of the Dryad digital repository, represents an open-licensed educational module which has been developed by collaborations with researchers and educators for students of all levels (e.g. secondary, undergraduate and graduate) to make use of real data in their work. Furthermore, figshare [81], a website conceived by Mark Hehnel, is a platform whereby scientists can deposit all of their data which, when uploaded, is given a citable digital object identifier (DOI) based on the Handle System thereby ensuring efficient searches and security of the stored data for long-term access. Therefore, the massive amount of data accumulated through scientific research activities that never get published, can be shared. This in turn could drastically reduce the expenses involved with the attempt to duplicate experiments [82]. Moreover, Figshare also encourages the deposition of data that has been generated but never published. Similarly, Zenodo [83] is an open research repository developed by OpenAIRE and CERN in 2013 based on the Invenio digital library framework which also supports DOI versioning for researchers of all fields. Additionally, the reporting of research funded by the European Commission via OpenAIRE is also integrated into Zenodo whereby all research is stored in the cloud. Furthermore, the Open Science Framework (OSF) [84] is a cloud-based tool promoting the open and centralized management of scientific workflows at all stages of the research process, with integrations from many other data hosting/repository services (e.g. Dropbox, GitHub, Google Drive, figshare, etc.). The OSF was developed in 2013 by a non-profit organization known as the Center for Open Sciences (COS) [85] for conducting research that supports and builds the scientific community by promoting the reproducibility and integrity of research. In addition, OSF not only supports researchers in the scientific community, but also software developers and publishers allowing for institutions to create and manage projects which can be shared via posters and presentations in meetings and conferences [86]. Additionally, publishers are further facilitating data sharing by establishing data journals that allow researchers to share their data in publication format that comes equipped with citable bibliographic details and DOI (i.e. without the need to provide full analysis that is typical of full-length research articles). Notable examples include Nature's Scientific Data [87], Elsevier's Data in Brief [88], MDPI's Data [89] as well as F1000Research [90]. It is also worthy to note that pre-prints also represents an important source for disseminating not only data papers but also full-length research articles while the actual manuscript may be under the peer-review process. Notable pre-print journals include arXiv [91], bioRxiv [92], ChemRxiv [93] and PeerJ Preprints [94].

Code repositories act as an archive for file and web facilities which are deposited either publicly or privately. Most often, they are used by open-source software projects as they allow developers to submit organized patches of code into the repositories supporting version control. Some of the main examples include GitHub and BitBucket. GitHub is a web-based, distributed version control system that allows developers to collaborate with people anywhere in the world, using a single codebase on the GitHub web interface. Both small and large projects can be handled with speed and efficiency using GitHub. Similarly, BitBucket [95] is a Git and Mercurial code management and collaboration platform used by professional teams to build, test and deploy software. A special feature included in BitBucket known as pull requests, allows code review that results in a higher quality of the code produced which can also be shared amongst the team. In addition, branch permissions in the BitBucket software provide access control thereby ensuring that only the right people are able to make changes to the code. It should be noted that not all codes and data can be made publicly available and in such circumstances, private repositories such as GitLab [96] represents a lucrative solution. Furthermore, cloud-hosted source repositories such as Assembla and Google Cloud Platform, provide storage facilities where the code can be kept secure without the threat of a hardware collapse. Assembla [97] represents the only multi-repository provider used for hosting repositories (e.g. Subversion, Git and Perforce) in the cloud that also answers the requirements for compliance. Assembla also provides the use of cross-platform applications which can integrate seamlessly with other modern cloud services such as JIRA, Jenkins and Slack. The Google Cloud Platform [98] on the other hand, utilize Git version control for supporting collaborations of various applications, including those running on the App Engine (i.e. a cloud computing platform for hosting web applications) and the Compute Engine (i.e. the service component of the Google Cloud Platform). Google Cloud also provides a source browser through which, the viewing of uploaded repository files is possible. Moreover, it is able to integrate source code already present on GitHub or BitBucket seamlessly onto its cloud platform.

Interactive code platforms such as Binder and Code Ocean, allow subscribers to collaborate in real-time via a remotely hosted web server without the need for software installations. Binder [27] allow subscribers to deposit their GitHub repositories containing the Jupyter notebook via a URL, which is then used to build a Binder repository. Dependency files from the uploaded environment generates a Docker image of the repository. Furthermore, a JupyterHub server hosts repository contents thereby allowing easy access to live environments as well

as facilitate sharing with others using a reusable URL. An article by Sofroniew et al. [99] on neural coding published in eLife made use of Binder to share data on all neural recordings. Furthermore, an article published in Nature by Li et al. [100] on the robustness of neural circuits, used the Binder platform to share their computational simulation results. In addition, Code Ocean [101] is a cloud-based computational platform that allows users to share and run codes online, thus encouraging reproducibility. Partnership between Code Ocean and publishers would further encourage reproducibility in which readers can gain access to the executable algorithms right from the published articles as is the case for IEEE journals [102].

Taken together the aforementioned resources supports the hosting, using and sharing of data/code, which sets the stage for the paper of the future as also discussed by C. Titus Brown in a Nature TechBlog [103]. Brown also suggests that non-technophiles can learn the tools of the trade that facilitates computational reproducibility by attending training workshops such as those provided by Software Carpentry [104] and Data Carpentry [105].

Model development in computational drug discovery

In silico models can be generated to study a wide array of chemical and biochemical phenomenon. In this section, we consider aspects of the model building process and key issues on what is needed to generate sufficiently accurate, reproducible models. We consider the full range of computational models ranging from ligand-based cheminformatic methods and molecular mechanics-based models to high-level structure-based simulations involving protein-ligand docking or protein-substrate reactivity.

In the context of computational drug discovery, it is in our view that it is important to differentiate between the reproducibility and the predictability of a computational model. The former relates to how accurately subsequent predictions on the same compounds change over time when compared to that proposed using the originally developed computational model. Alternatively, reproducibility could also relate to the model building procedure, whether the data was sufficiently sampled such that repeating the process would not lead to dramatically different results. In most cases, differences can occur between different models that depends on how the data sets were sampled and how the models were generated, implemented or maintained. For example, deviations can occur if: (1) an unrepresented data set are selected for model building, (2) following implementation a different variant of a particular descriptor engine is used for the model (i.e. clogP, AlogP, etc.) or (3) during the production phase, descriptor coefficients are truncated or rounded off. These sources of reproducibility errors can of course be easily monitored by periodically re-running models on the original data sets and descriptors. This naturally brings us to the intrinsic accuracy of the model itself, which is related to how well our theoretical description of the phenomenon (i.e. being models) can actually describe the physical event taking place. *In silico* models cannot describe an experimental event with the same level of accuracy than performing the experiment a second time. Thus, small differences in the reproducibility of a given model may not make a significant difference to the utility of the method in the real world. Furthermore, models that have what might be considered to be a finite predictivity (r^2 of 0.5) can indeed be useful in drug discovery. In these cases, updates to particular descriptors engines at the back-end of a prediction tool will in all probability have negligible impact on the overall model statistics if generated on a large, diverse data set. Thus, while the absolute predictions for each compounds will change, the overall effect on the accuracy of the prediction is likely to be negligible. A key issue is therefore to build an intelligent system or workflow such that rounding errors or other subtle differences can be differentiated from the more serious algorithmic or implementation errors.

Chemical and biological data repositories

The implementation of open data initiatives by many fields including bioinformatics and proteomics has dramatically risen in the past few years with the Human Genome Project being paramount in guiding the scientific community towards open science [106]. However, researchers in the pharmaceutical industries lack the appropriate informatics knowledge that would allow them to completely make use of such platforms. Hence, the availability of cheminformatic tools that are easy to use can help reduce time and cost in this complex drug discovery field [107]. Similarly, various connections between protein and ligands can also be established using these widely available resources [108]. The presence of a large number of experimental and biological databases containing relevant screened compounds is easily accessible via a public domain. Among them, the most widely used databases are ChEMBL [109] and PubChem [110]. The ChEMBL bioactivity database is a large open-access drug discovery database comprised of more than 2.2 million freely available compounds obtained from over 1.8 million assays having around 15 million activity values [109]. In a similar fashion, PubChem was established by the National Center for Biotechnology Information (NCBI) as a public repository that gathers information on biological activities of small molecules. In addition, PubChem currently contains a database of about 96.5

million compounds with bioactivities for greater than 237 million [110].

Furthermore, another publicly accessible database known as Binding Database or simply BindingDB [111] contains experimental small molecules interaction data from patents and scientific articles making up more than 1.4 million protein-small molecule affinities with over 7,000 proteins involved with greater than 650,000 small molecules as of January 2019 [112]. In addition, DrugCentral [113] and DrugBank [114] are comprehensive resources focusing on FDA approved drug that combines the chemical, pharmacological and pharmaceutical information of the drug with the sequence, structure and pathway information of its target. This latest update of DrugBank [115], shows a tremendous increase in drug-drug interaction data for ADMET properties as well as additional new features such as pharmaco-omics data with special focus on pre-clinical and clinical trials. These additions and enhancements are intended to facilitate research in pharmacogenomics, pharmacoproteomics, pharmacotranscriptomics, pharmacometabolomics, pharmacokinetics, pharmacodynamics, pharmaceutics and drug design and discovery. In addition, databases such as CARLSBAD [116], BRENDA [117] and ExCAPE-DB [118] contain uniformly presented data integrated and curated from various repositories .

Ligand-based approaches

Ligand-based drug design is based on identifying key features that give rise to biological activity and aiming to incorporate, improve or identify new chemotypes with similar characteristics. Pharmacophore-based models are based on 2D or 3D methods and assume that all molecules that contribute to said pharmacophore bind in a similar manner to the prospective target [119–121]. Such similarity may be used to identify compounds with the same or similar features, or can be employed in conjunction with statistical methods to give either structure-activity relationships (SAR) or the more extensive QSAR. Such methods are useful to ascertain trends within primary screening data. The intuition of the medicinal chemist is critical at the beginning of a project, however the large amount of early screening data generated mean that visual analysis of the data is not practical [122]. Thus, compounds clustering and SAR analyzes can provide a simple, efficient means to explore or generate initial SAR. This allows one to identify a series of molecules with the greatest potential and develop new molecules with relatively localized structural changes to significantly assess the activity landscape.

In the context of chemical risk assessment, toxicological profiles of chemicals (cosmetics, industrial chemicals, food chemicals, etc.) are often tested in animals prior to human consumption or usage. QSAR has been proposed as a promising replacement to animal testing [123] or if experimental testing is inevitable then QSAR can help to (1) supplement experimental data and (2) prioritize chemicals for such experiments [124, 125]. Particularly, QSAR can help in regulatory purposes as it can be used to generate structural alerts or *"expert rules"* derived from SAR observations that relates a structural template or functional group to a particular adverse event (i.e. toxicity and undesirable pharmacokinetic properties). As the name suggests, the rules can be a result of expert intuition or from statistical analysis of representative data sets. Many examples exist, including for DNA reactivity [126], toxicity [127, 128], skin sensitivity [129, 130], pan assay interference (PAINs) compounds and general purpose filters for undesirable compounds [131–133]. An alert does not mean that a toxic event is to be expected per se, rather it acts as a qualitative prediction of increased risk. This means that such models can be used for guidance purposes only. Indeed, Alves et al. [134] noted the concern that these structural alerts can disproportionally flag too many chemicals as toxic, which questions their reliability as qualitative markers. The authors state that the simple presence of structural alerts in a chemical, irrespective of the derivation method, should be perceived only as hypotheses of possible toxicological effect.

QSAR modeling involves generating multivariate predictive models using chemically relevant descriptors (e.g. structural counts, fingerprints, 2D and 3D molecular properties, etc.) along with biological activities [135–140]. There are thousands of potential molecular descriptors of numerous types that can be used to explore the complex relationship between structure and response. In such cases, care needs to be taken as the probability of finding spurious correlations, particularly with small data sets, is significant [141, 142]. Biological responses (e.g. inhibitory activity) will typically undergo logarithmic transformation or be used to define subclasses to facilitate the statistical model building process [143]. Model building can be performed using a wide variety of methods ranging from simple statistical methods (e.g. multiple linear regression) to machine learning methods (e.g. random forest, artificial neural network, support vector machine).

Model performance then needs to be assessed using a wide range of statistics, including correlation coefficients, estimates of prediction error, etc. For classification models, false positive and negative rates as well as holistic measures such as the Kappa statistic or the Matthew's correlation coefficient are recommended [144, 145]. To understand the true predictive capability of the model it can be instructive to look at how the errors or correlation compare with repeat measurement from the

experimental assay being modelled. QSAR models cannot be more predictive than the data they are built on. If such a situation is encounter, it would suggest the model is overfitted and may not extrapolate well to future compounds [143]. In such cases, additionally statistical validation in the form of leave one, or leave many out cross validation, or Y-randomization trials can be useful [142, 145].

Aside from assessing the performance of constructed QSAR models, current efforts are already in place for establishing the reliability of QSAR models, for instance, by using conformal predictions [146, 147] and applicability domain [143], which have been proposed as promising approaches for tackling this issue. Putting this into perspective, the statistical performance as produced by conventional metrics such as R^2 or RMSE suggests how well the model is performing on the prediction task but it does not consider whether such predictions are made on compounds falling within the boundaries of the applicability domain or the degree of certainty that the model has on the predicted bioactivity of compounds. Such confidence and the degree at which a compound falls within the applicability domain would greatly assist in compound prioritization. A further practical look into applicability domain will be discussed in the forthcoming paragraphs.

QSAR models can only be as good as the data that they are built on therefore, it is to be expected that they would not be able to predict as good as repeating the experiment a second time. QSAR models are highly useful as a first filter however, users and developers face a number of issues while generating and using the models in practice [148]. The quality of the models can be evaluated with statistical parameters, including correlation coefficient or root mean square error (RMSE). It is expected that the RMSE cannot be smaller than the RMS of the experimental method, otherwise the model is overfitted [149]. Many data sets that require investigation consist of diverse compounds sets of finite sizes (i.e. commonly 100-10000 in size) and are sometimes termed global models [150, 151]. This means that any model built could easily be overfitted due to the typically small number of observations and large number of descriptors and modeling methods. Another issue is that any new compounds may not be very similar to those used to build the model and may therefore be poorly predicted, which could occur even if the model is apparently highly predictive. In that case, the distance of the compounds to the training set model space can be used to estimate the probability of the prediction reliability.

Generally speaking, a global model built on a large diverse data set would be expected to generate a better prediction on an unknown compounds compare to one generated on a small set of compounds. However, in some cases QSAR models built on small congeneric series can be highly useful when restricted to the chemotype in question. This is particularly true if these related molecules act via a similar mode of action so that the activity to be explained is affected by fewer factors. These so called local QSAR models are built using only a specific class of chemotypes, and have a limited domain of applicability [150, 152]. However, they are often more predictive for the subset of chemicals that they can be applied on specifically because of the fewer confounding factors contributing to the activity. Additionally, in terms of interpretation, local models can be more useful in a practical sense because it is possible to understand what the models are telling in order to obtain new molecular insights from the model. Therefore, when a novel compounds under investigation have a common structural cores, a medicinal chemist could carefully choose chemist friendly descriptors not only to get a robust model but also provide useful information on which descriptors that modulate biological activity. Local model can provide useful information to the medicinal chemist on how to improve biological activity by linking the descriptors (e.g. lipophilicity, electron donating or withdrawing properties, hydrogen bonding effects and molecular size).

The domain of applicability of QSAR models can be used to give the user a degree of confidence in the prediction as it can be shown that there is often a correlation between the query compounds and those in the training set as calculated using the model descriptors [143, 152]. It is expected that compounds that lie within this reason should be better predicted than those that lie outside, assuming of course that the model is not overfitted. The compounds from the test set, if reasonably similar to those from training set (i.e. due to selection procedure, or due to random sampling), then the model should perform well. However, if more challenging validation sets are chosen, such as unrelated compounds, or compounds prepared at a later date which typically show greater dissimilarity, the statistics are generally less favourable [143]. However, if the test chemical is far away from the training set a valid prediction cannot be expected. Once the model is established, then one can make a prediction and consider its reliability. To inform the quantity of the information available to the model, information towards a query structure can be obtained by averaging distance (e.g. Tanimoto, Euclidean, etc.) between the nearest neighbors. The reliability of the information that is in the model for a given prediction is normalized into 0 to 1 range in which 0 has the nearest distance and 1 the farthest distance. If the query compounds are in the AD of the model and the prediction is in the reliability

domain, then the prediction can be concluded as valid and reliable [143].

Guidelines on the development of robust QSAR models based on the Organization of Economic Cooperation and Development (OECD) principles of validation have already been published [153]. With recent emphasis being placed on the reproducibility of models, Judson et al. [154] proposed the Good Computer Modelling Practice (GCMP) guidelines which identifies the best practice for conducting and recording modelling procedures. Although, with the availability of ample literature on the best practice in QSAR modeling [155], it is mostly aimed at those having cheminformatics/mathematical understanding of the subject. In a recently published article, Patel et al. [156] assessed the reproducibility of QSAR models pertaining to ADME predictions by scientists without expertise in QSAR. The authors reviewed 85 papers spanning 80 models with ADME related endpoints and presented a pragmatic workflow for the implementation of QSAR models with greater usability. In addition, QSAR models are able to correlate the physicochemical properties of a structure with the biological activity [157]. Hence, the QSAR models can be efficiently used for the activity prediction of unknown compounds and designing new compounds for that particular activity. However, many of the QSAR models published, are not aimed at drug design. In that regard, Kurdekar and Jadhav [158] designed an open source Python script for QSAR model building and validation using data for Matrix Metallo-Protease 13 (MMP13) inhibitors and a series of anti-malarial compounds.

Structure-based approaches

Structure-based computational approaches generally require greater input from the model builder, resulting in a larger number of approximations being used to generate the predictive model. For example, which protein crystal structure do we choose for a particular target, how do we treat ionizable residues, are all residues flexible, what method and parameters will we use to model the results, what software program and custom parameters will we use etc. This results in a large amount of often subjective decision being made. However, while these results may change the outcome of the simulations, it is hoped that it will not impact on the overall conclusions.

For example, there are numerous protein structures that are available from multiple families via the Protein Data Bank (PDB) [159] therefore it is possible to either obtain high quality structures of the target of interest or generate homology models [160] for use in computational analysis. One of the most commonly employed technique is molecular docking, a technique that samples and scores conformations of small molecules bound to a target active site [161, 162]. Docking and scoring algorithms are employed to predict protein-bound conformation, virtual screening of large data sets and sometimes to try and estimate molecule potency.

The information gained from docking exercises can be invaluable for helping rationalize SAR and help inspire further synthetic plans. However, care should be taken as to not over interpret such models. A detailed study by Warren et al. [163] shows that although docking could successfully predict the protein-bound conformation and explore conformational space to generate corrected post as well as correctly identify molecule or chemotypes of actives from a population of decoy molecules, they are less successful in identifying the post closest to the crystal conformations using scoring functions and that no single docking protein performed well across multiple protein targets. The inconsistencies of the docking program to reproduce greater than 35% of the binding modes within 2 Å across all targets highlight the fact that experts in the loop, or additional experimental data is often needed to correctly predict binding modes.

QSAR can also be applied to 3D models of ligands that are (a) superimposed together based on a common active conformation or (b) superimposed based on how they dock within a given active site. All the complications that are pertinent to the generation of 2D QSAR models also apply to 3D QSAR. However, there are additional issues that can arise in 3D QSAR models due to the extra assumptions that must be made: (1) receptor binding is related to the biological activity (2) molecule with common structures generally bind the same way, (3) the properties that govern the observed biological response are determined by non-bonding forces, (4) the lowest energy conformation of compound is its bioactive conformation and (5) all the ligands in the study bind the same target site and have a comparable binding mode or similar mode of action [164, 165]. To perform a 3D QSAR, the chemical structures are optimized using molecular mechanics, or to a lesser extent semi-empirical and quantum mechanics to obtain a lowest conformer. These can then be overlaid on a common ligand scaffold and these coordinates are used to determine descriptors (e.g. CoMFA and CoMSIA). Chemical structures can also be superimposed using docking to a target binding site, using field based or pharmacophore-based methods. The whole process is predicated on the fact that modelled 3D structure corresponds to the active conformation. A further limitation of such models is that, compound hydrophobicity is not so well quantified, and many descriptors are produced, most of which have low variance [166].

3D models can be further expanded with additional approximations. For example, molecular dynamics simulations that are based on empirical MM parameters can

be applied to simulate how molecular systems evolve over time using Newtonian mechanics. These simulations are based on rather simple molecular methods (e.g. AMBER, CHARMM parameters, etc.) which is necessary to obtain sufficient conformational sampling. Nevertheless, computational sampling could be sacrificed if the user wanted to use more accurate quantum mechanical methods, starting with semi empirical treatments such as AM1 or PM3, to ab initio methods such as Hartree Fock (HF) to methods that take into account electronic methods such as DFT (e.g. B3LYP and M0X series) [167]. A further advantage of the latter methods is that they do not require bespoke generation of parameters for each molecule. However, the massive overhead in computation means they are rarely used for protein simulations. Alternative methods to get around the flaws of both methods include hybrid quantum mechanical/molecular mechanical methods. In these methods the active site region that contains the substrate/inhibitor and the main residues it interacts with are defined using QM, and the remaining protein MM. This method makes it possible to perform more accurate evaluation of the energetics while also providing a means to perform molecular dynamics over acceptable time frames [168–172]. Despite these advances, a suitable balance between methods of sufficient accuracy, and sampling of sufficient time eludes us.

Systems-based approaches

Systems-based drug discovery aims takes a holistic view of the genome, proteome and their specific interactions amongst one another and how chemicals may postively or negatively modulate their action [173, 174]. Particularly, this encompasses the understanding of the underpinning details of biochemical pathways in which the interplay of gene, proteins, carbohydrates, lipids and chemicals sustain the molecular logic of life. As there are more than 30,000 genes that may subsequently translate to proteins via complex gene expression feedback loop, therefore such vast amounts of data requires the utilization of computers for extracting key insights. Systems biology take a broader overlook of biological systems as oppose to the convention reductionist approach. The field pieces together disparate information from various omics disciplines to produce a unified analysis of the data.

In the context of drug discovery, systems pharmacology (i.e. also termed "network pharmacology") makes it possible to perform drug repositioning [175, 176] in which known FDA-approved drugs that were originally designed to treat disease A (i.e. original indication) can be repurposed or repositioned to treat other diseases (i.e. new indication). This is made possible owing to the concept of polypharmacology that essentially relies on the concept of molecular similarity whereby similar target proteins are assumed to also share similar binding characteristics to compounds [177]. For instance, the nelfinavir (i.e. an HIV-1 protease inhibitor) has been demonstrated to exert promising anti-cancer activities against a wide range of cancer types [178]. Aside from network pharmacology proteochemometric modeling is systems-based approach that has also been demonstrated to facilitate drug repositioning [179, 180].

Hereafter, we examine the ongoing work in the effort to establish reproducibility of systems biology models. The COmputational Modeling in BIology NEtwork (COMBINE) is an initiative that has been set up in 2010 to coordinate the development of various community standards and formats pertaining to the development of systems biology models. Two independent articles published in the *IEEE Transactions on Biomedical Engineering* examines this topic in which Waltemath and Wolkenhauer [181] focused on how initiatives, standards and software tools supports the reproducibility of simulation studies while Medley et al. [182] formulated a set of guidelines for building reproducible systems biology models. Moreover, Waltemath et al. [183] outlined the necessary steps needed to facilitate the production of reproducible models in the systems biology setting by exemplifying a number of computational models pertaining to the cell cycle as obtained from the BioModels database [184]. The authors summarized that in order for models to be reproducible, they should be (1) encoded in standard formats (e.g. XML, SBML, CellML, etc.), (2) the meta-information should be provided to support the understanding of the model's intention, (3) associated simulation experiments should be encoded in standard formats and (4) all information must be made available through open repositories.

Kirouac et al. [185] investigated the reproducibility of quantitative systems pharmacology (QSP) by analyzing 18 QSP models published in the *CPT: Pharmacometrics and Systems Pharmacology* journal. Owing to the heterogeneity of the platform used in the 18 models, 12 were selected for further analysis (i.e. coded in R, PK-Sim/MoBi, and MATLAB) and only 4 were found to be readily executable from a single run script. From there, the authors initiated points for discussion on how to establish best practices for QSP model reproducibility. Notable points raised includes: (1) suggesting the provision of a single run script to allow interested users to easily perform the simulation, (2) journals should provide recommendations on the sharing of code and data, (3) provide sufficient details on the setup of the simulation model, (4) provide models in open source, standardized format, (5) provide details on the computation environment (e.g. software version, parameter details, etc.).

Watanabe et al. [186] discussed the challenges of disease model reproducibility that had predominantly relied on periodically evolving loose guidelines as opposed to well-defined machine-readable standards. Thus, the authors investigated the utility of Systems Biology Markup Language (SBML) [187] in the development of disease models as compared to other associated languages including Pharmacometric Markup Langugae (PharmML)[188] and MIcro Simulation Tool (MIST) [189]. Results indicated the robustness of SBML for model reproducibility and as the authors pointed out, there exists substantial adoption of SBML where most are being deposited to the BioModels repository [184].

In addition to aforementioned markup languages for facilitating the exchange of models, there also exists other languages as well such as CellML[190] , Simulation Experiment Description Markup Language (SED-ML) and Systems Biology Graphical Notation (SBGN). In the presence of these various markup languages, the research group of Sauro [191, 192] proposed the Tellurium platform as an integrated environment (i.e. models, Python code and documentation; similar to a Jupyter notebook) that is designed for model building, analysis, simulation and reproducibility in systems biology while facilitating the use of multiple, heterogeneous libraries, plugins as well as specialized modules/methods. Similarly, BioUML[193] is a web-based, integrated platform that facilitates the analysis of omics data in the context of systems biology. Furthermore, the extensive collection of the plug-in architecture (i.e. comprising more than 300 data analytic methodologies coupled with its ability to integrate with the Galaxy and R/Bioconductor platforms) positions BioUML as a prominent platform for building systems biology models. Moreover, a workflow engine integrated into the BioUML helps to support the concept of reproducible research as new input data can be plugged into the already existing model pipeline. Additionally, Drawert et al. [194] developed MOLNs as a cloud appliance that entails setting up, starting and managing a virtual platform for scalable, distributed computational experiments using (spatial) stochastic simulation software (e.g. PyURDME).

Computational issues on model development and deployment

There are two main issues facing the computational scientist or model developer in drug development: computational processability and scalability. Irregardless of where computation is performed (i.e. on a laptop, a server, a data center or a cloud infrastructure) in order to achieve the reproducibility in sufficient details, it is crucial that tools for structuring and managing these processes are implemented and exploited as drug research pertains to many activities involving various data types of different sizes and formats. In a typical computational drug discovery project, it becomes very difficult to keep track of tools and parameters that were used, the different versions of data as well as the manual gluing of results together into the final tables and figures that are presented in a scientific manuscript. Proper data management becomes a key necessity. Challenges of data management in the big data era have previously been discussed [195] and practical suggestions on how to structure data in computational analysis projects have also been proposed [196]. Furthermore, as soon as a data set increases in size above a few tens or hundreds of gigabytes, or when the amount of data needed to be kept in RAM becomes larger than a few gigabytes, it often becomes infeasible to perform computations on the user's own local laptop, and therefore scaling up the computation on a larger computing infrastructure becomes an inevitable need. In this section, we discuss the most common approaches for resolving these challenges.

Scientific workflow management systems

For reproducible research, an important capability is to be able to re-run and validate a complete analysis pipeline in an automated fashion. While this can, to some extent, be done by scripting, scripted pipelines can easily become brittle and complex to manage and modify due to their low-level nature. As the user is forced to take care of all the low-level details of data management and program execution, even simple changes in the workflow can require a substantial mental effort in order not to introduce subtle errors. Also, tracking intermediate output from an intermittent process of the pipeline can be difficult in order to determine which process causes the failure. Optimizing which step to be rerun instead of restarting the whole pipeline is an important question in the context of large-scale analyses. These problems are at the core of what scientific workflow systems aim to solve, and thereby contributing to making computation research more reproducible.

Scientific Workflow Management Systems (WMS) provide a number of added benefits to computational pipelines that help in creating reproducible, transparent computations. Firstly they allow the user to construct the pipeline using a more high-level, abstract description than plain scripts, hiding away low level technical details of exactly how data is managed and programs are executed. The user typically only needs to specify how the computational steps depend on each other, and which parameters to feed them with. The WMS takes care of low-level details such as scheduling the concrete

invocations of the workflow steps in the right order with the right parameters, passing on data between processing steps, separating unfinished and finished files (in what is often called atomic writes), logging, producing audit reports and more.

The more high level description of workflows in WMS, primarily consisting of task and data dependencies, makes it easier to follow the logic of the core computation making up the pipeline. It also makes the workflows easier to change as one typically needs to change the workflow code in far fewer places than in scripts, because of the lower amount of details specified in the high-level description.

In summary, WMS benefit reproducible computations by (1) making automation of multi-step computations easier to create and more robust and easy to change (2) providing more transparency to what the pipeline does through its more high-level workflow description and better reporting and visualization facilities, and (3) by providing a more reliable mechanism for separating unfinished and completed outputs from the workflow.

The most common WMS used in drug discovery over the last few decades have been the proprietary Pipeline Pilot software [197] and the open source KNIME workbench [198], which also has proprietary extensions. Over the years, the relative use of KNIME appears to have increased. This is illustrated by the distribution of articles available on PubMed which mention *"Pipeline Pilot"* or *"KNIME"* in their title or abstract (Fig. 3). It should be noted that a PubMed search might not provide the full picture of the usage of these software inside for example pharmaceutical industry, since PubMed only indexes public, peer-reviewed articles, the production of which is not the primary concern for most pharmaceutical companies. A part of the increased number of papers mentioning KNIME might also be due to the generic nature of the platform, as KNIME is not focusing only on drug discovery, but also has support for more general data analysis tools, as well as tools specific to other fields even outside of the biomedical domain.

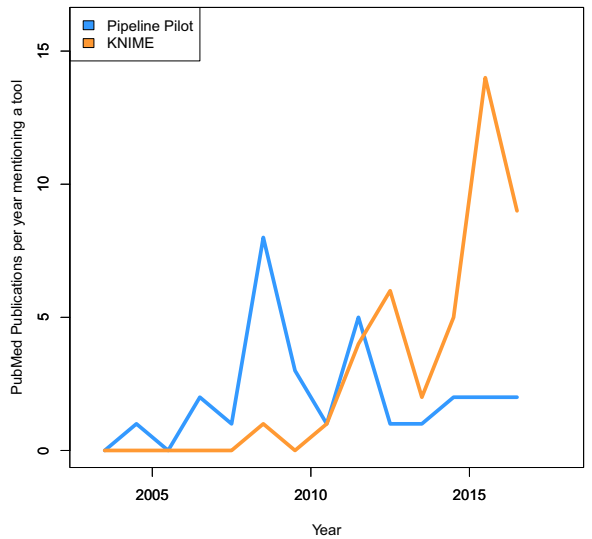

Fig. 3 Number of articles on PubMed, mentioning *"Pipeline Pilot"* or *"KNIME"* in their title or abstract from 2003 to 2017

Pipeline Pilot has been used in several studies, including to design screening libraries [197], for high-content screening [198], and for compound design [199]. KNIME has been used, for example, for virtual screening [200], target identification [201]; more in-depth coverage of applications are provided elsewhere [202].

In addition to Pipeline Pilot and KNIME, there has been some use of the Taverna and Galaxy platforms too. Taverna, which has been widely used in the wider bioinformatics field in the past, has functionality relevant to drug discovery through the CDK-Taverna project [203], which integrates the JVM-based Chemistry Development Kit [204, 205]. The immensely popular web based Galaxy platform [49–51] has the ChemicalToolBoX, which is a suite of more than 30 tools for chemistry and cheminformatics integrated [206].

A recent trend among many more recent workflow tools popular in bioinformatics, is that the main mode of interaction with the user is increasingly often purely text-based. Prominent examples of this trends include tools like Nextflow [207], Snakemake [208], Ruffus [209], BPipe [210], Cuneiform [211] and Luigi [212]. Discussions with users of workflow tools reveals that this focus has a lot to do with the easier integration of workflows into HPC and cloud computing environments as well as easier version control when all workflows are stored as plain text files rather than as configurations in a GUI software. Keeping track of all changes and versions to workflows in version control is identified as one key component in achieving reproducibility in computational biology [213, 214].

Among these newer text-based tools, Luigi has found some use in drug discovery. The fact that Luigi is implemented as a Python library, enables it to seamlessly integrate with python based client programming libraries such as the ChEMBL client library [215]. By not requiring a GUI, Luigi is also easier to integrate and run in an HPC environment, interacting with resource managers such as SLURM. This was recently done in a study on the effects on dataset and model sizes on the predictive performance of toxicity models [216]. SciLuigi [217] is a wrapper library around Luigi, designed specifically to make workflow motifs common in drug discovery easier to model with Luigi. An example of of such motifs are

machine learning pipelines containing cross-validation of trained models, nested with parameter sweeps. SciLuigi also includes built-in support for the SLURM HPC resource manager [218].

Another trend in the wider field of computational biology is increasing adoption of support for tool-agnostic, interoperable workflow description formats such as the Common Workflow Language [219] or Workflow Description Language [220]. Such tool-agnostic formats promise to make it easier to share workflows with other users, who might prefer or even be restricted to, other tools and infrastructures, and can thereby make reproduction of computational studies easier. Use of such interoperable formats has yet to see widespread use within drug discovery, but presents a promising direction for increasing the reproducibility of computational studies in the field. By being a textual representation of workflows, they may also provide an excellent way for GUI-centric workflow systems to provide a representation of its workflows that fits in easily with popular version control systems like Git.

Large-scale integrative computational infrastructure

High performance computing (HPC) clusters

The traditional way of scaling up scientific computing workloads has been by using high performance clusters. These have in the last couple of decades typically consisted of so called Beowulf clusters, meaning clusters composed of relatively *"normal"* computers, running a common operating system such as Linux, and connected through a high performance network. These compute nodes typically mainly only differ from normal computers by possibly having more compute cores and/or random access memory (RAM). Workloads on HPC clusters can either run within one node, much like any other program, or use a technology such as Message Passing Interface (MPI) to run a computation by running the program on multiple nodes, where the multiple instances communicate with each other via MPI. The latter is a common scenario in physics, but is not widespread for computations in the biomedical field.

Despite of the recent trend towards cloud computing environments, HPC still remains a common option especially for academic computing because of the relatively low cost per CPU hour. On the other hand, HPC environments typically do not allow the same level of flexibility and user control as cloud environments, because of tighter security requirements, and various policies induced by local system administrators. For example, it is typically out of question to get root privileges on a HPC compute node, or to install your own virtual machine, where you could get root privileges. This means users sometimes need to compile and/or install the required software by hand, if the right version of the software they need is not already available on the cluster. There are some recent trends to meet the need for software packaged into container, most notably through the Singularity project, which allows users to run a type of container without root privileges.

Cloud computing and virtualization

Cloud computing offers computational infrastructure, platforms, and services on-demand, and it will have a profound impact on how computational drug discovery is carried out [221, 222]. For pharmaceutical companies, on short term perhaps the highest impact is the on-demand availability of computational infrastructure, relieving them of the burden to manage an in-house computing center. But in the longer run, platforms-as-a-service supporting drug discovery has the potential to dramatically change the way computer-aided drug discovery is carried out, for example, accelerate processes [223] and scaling up analyses [224], but also at the same time drastically improve reproducibility.

Virtual machines

Some software tools and workflows/pipelines can be complex to move between systems, even if they are open source and all data is publicly available. For example, when installing the same software on different systems, there will always be different versions in some dependent packages and different optimization flags for compilations etc. that could affect the execution of software and lead to different results in analysis [207]. One way of addressing this problem is by using virtual resources. A virtual machine (VM) is an emulation of a computer system that provides functionality of a physical computer, with a complete operating system that runs within a managed *"virtual"* environment without direct connection to the underlying *"host"* computer. Virtual machines can be packaged as a virtual machine image (VMI or simply *"image"*) that can be transported between systems and launched on demand. In science, researchers can take a "snapshot" of their entire working environment including software, data, scripts etc that can be shared or published, and cited in publications to greatly improve reproducibility [225, 226].

VMs have been used in several drug discovery projects. For example, Jaghoori et al. [227] described how AutoDock Vina can be used for virtual screening using a virtual machine. McGuire et al. [228] developed 3d-e-Chem-VM, a virtual machine for structural cheminformatics research. Lampa et al. [217] provides a complete analysis using predictive modeling in drug discovery that is shared as a virtual machine image. Lilly has developed their Open Innovation Drug Discovery platform [229]

where participating investigators get access to tools and predictions by Lilly software and data via a virtual machine where they can, for example, submit compounds for in silico evaluation. The widely used ChEMBL database makes the data and tools available as a virtual machine via the myChEMBL package [230]. Virtual machines are also a necessity for Big Data frameworks in drug discovery, for example, implementing docking on Hadoop [231] and Apache Spark [232]. VMs can also be useful for providing student environments for educational courses, such as is done for the course Pharmaceutical Bioinformatics at Uppsala University [233]. There are several places to deposit virtual machines, for example, the BioImg.org website [234] is a catalog dedicated to housing virtual machine images pertaining to life science research. Further, VMIs can be shared within several public cloud providers (see Table 1).

Table 1 List of the largest public cloud infrastructure service providers

Service provider	URL
Amazon Web Service	http://aws.amazon.com/
Microsoft's Azure	http://azure.com/
Google Cloud Platform	http://cloud.google.com/
IBM's SoftLayer	http://www.softlayer.com/
Alibaba Cloud	https://www.alibabacloud.com/

Service providers are ordered according to market share [284]

Containers

A drawback of VMs to support computational reproducibility is that VMIs, with all software and raw data for an analysis available, tend to become rather large (i.e. in the order of several gigabytes). Software containers, or simply 'containers', are similar to virtual machines that they isolate software from its surroundings, but a container is smaller and do not contain the entire operating system; in fact, several containers can share the same operating system kernel making them more lightweight and use much less resources than virtual machines (Fig. 4). Containers can hence aid reproducible research in a way similar to virtual machines, in that they produce the same output irregardless of the system or environment it is executed on [226, 235, 236]. The most widely used containerization technology is Docker [70], but Singularity [237] and uDocker [238] are compelling alternatives that can run without root privileges and hence are more useful in shared high-performance computing facilities.

It is quite straightforward to containerize tools, and due to the portability it has become popular to ship tools for workflow environments such as Pipeline Pilot and KNIME [239]. However, containers in drug discovery is a relatively recent technology and not many published studies are available. Suhartanto et al. [240] presents a study for shifting from virtual machines to Docker containers for cloud-based drug discovery projects. The pharmaceutical company GSK describes in a presentation at DockerCon 2017 how they are able to accelerate

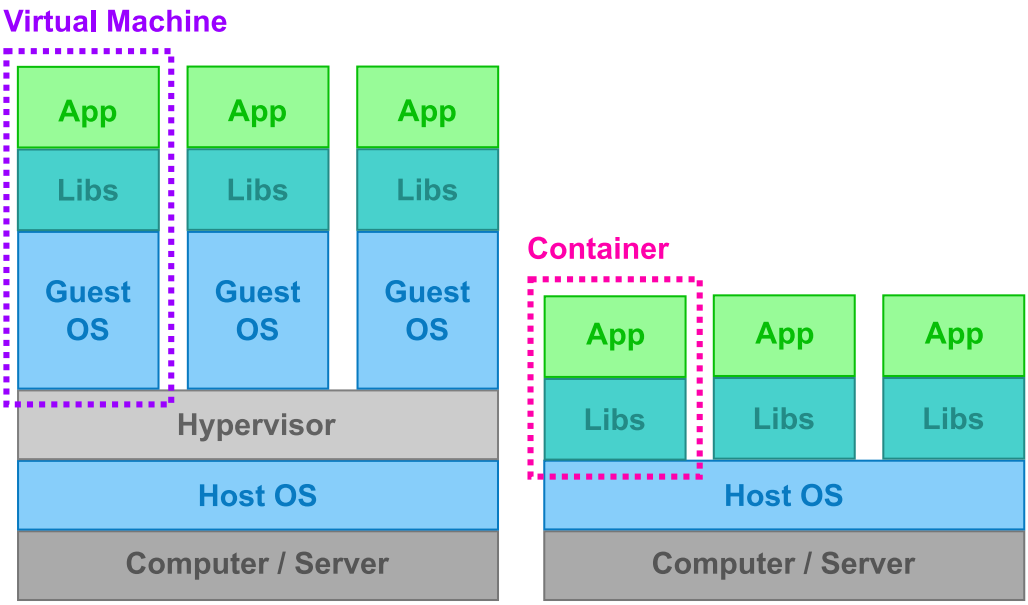

Fig. 4 Schematic comparison of virtual machines and containers. Virtual machines run on a Hypervisor and contains their own Guest Operating System. In contrast, Containers provide a layer of isolation that share the Host Operating System kernel and are hence smaller and faster to instantiate than virtual machines

science with Docker [241]. Altae-Tran et al. [242] applies Deep neural networks, available as a containerized version of their package DeepChem. Further, container technology is empowering e-infrastructures relevant for drug discovery, such as the OpenRiskNet project [243].

There are several repositories for containers, with Docker Hub being perhaps the most widely used. However, catalog services and standardization initiatives relevant for life science research also exist, with Bioboxes [244] and BioContainers [71] as two prominent examples. With the growing popularity of containers, it is very likely that we will see more virtualized tools, environments and studies become available using this technology in the future which will contribute to reproducible research.

Model deployment

Deploying a model in this context refers to installing it in a way so that it becomes accessible to oneself or others (Fig. 5). A model could, for example, be deployed on a laptop, a server on an internal network, on a private cloud for a selected group of people, or as a public service. Traditional model deployment as a service has commonly been done as a Web service available over a network, such as Internet. The service can then be accessed either via an HTML page that calls an application server that delivers results from the model, or via a Web API that can be consumed programmatically by software applications. There are some limitations of this simple model:

1. The service provider needs to maintain the service and the computer it runs on. If the service goes down, it should be restarted. Security patches must be applied. Hardware must be upgraded and replaced over time. This places a considerable burden on the service provider.
2. Whenever an update is made to the service, the version and possibly API will have to be changed. In order to sustain reproducibility, this soon leads to the maintenance of multiple versions on the same service.
3. If the service is resource-demanding, it can be expensive to offer it as a free service.

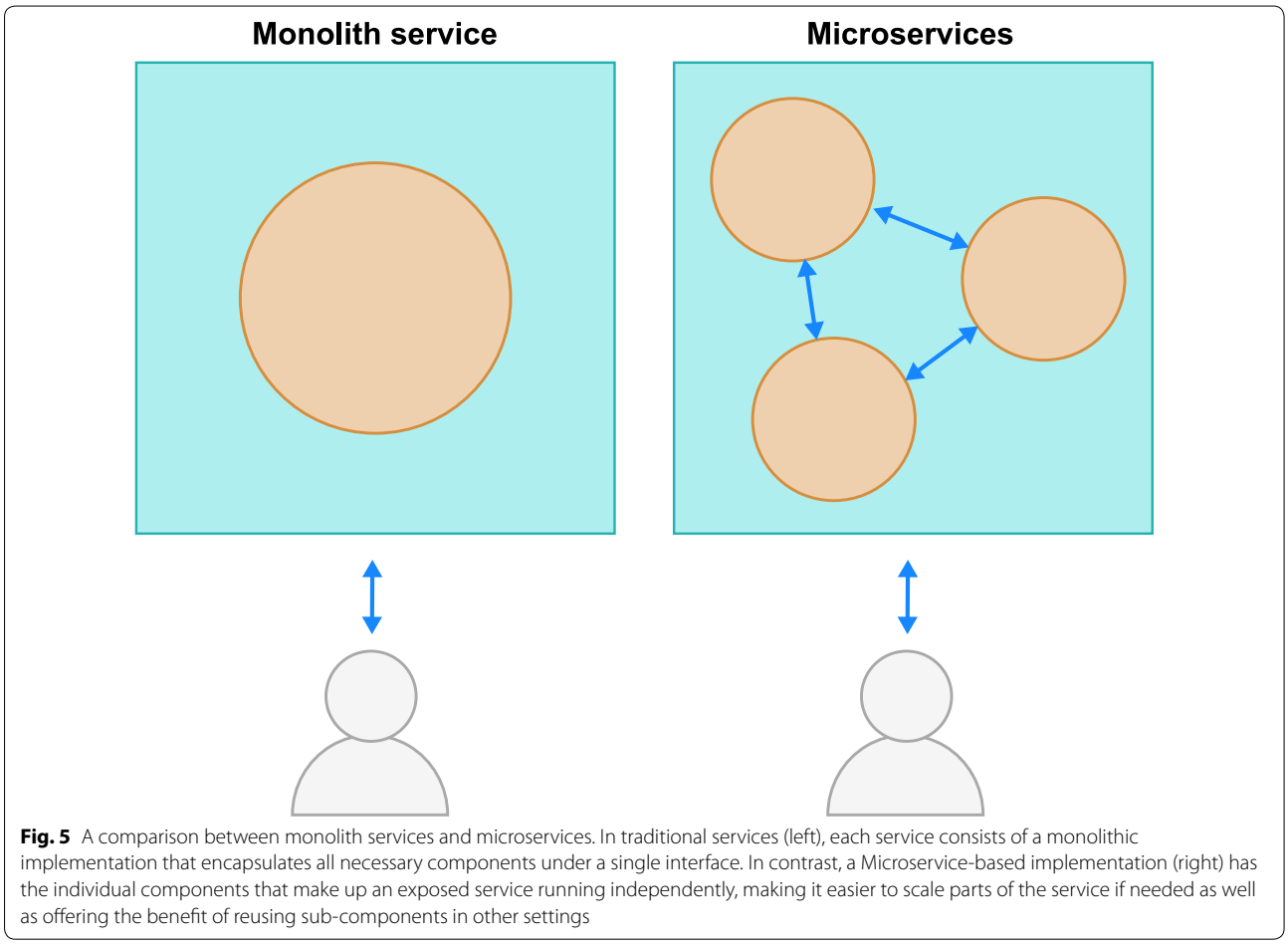

Fig. 5 A comparison between monolith services and microservices. In traditional services (left), each service consists of a monolithic implementation that encapsulates all necessary components under a single interface. In contrast, a Microservice-based implementation (right) has the individual components that make up an exposed service running independently, making it easier to scale parts of the service if needed as well as offering the benefit of reusing sub-components in other settings

These problems have limited the use of models deployed as services, apart from in-house services at companies with adequate system and service support.

Owing to the inherent complexities involved with setting up and maintaining fault-tolerant and scalable services, provisioning model services as virtual machines and containers has attracted a lot of interest [245]. Here it both becomes easier to publish a model online on, for instance, a cloud provider that eliminates the need to buy and maintain computational hardware, but also to enable users to instantiate the service on their own computational infrastructure. With proper versioning of services available (e.g. Docker containers) the end users can download and instantiate explicit versions of the model and ensure a reproducible component of an analysis. The problem becomes more how input and output data is structured, and there is a need for the community to develop and agree upon such standards for data, metadata including ontologies and vocabularies, and discoverability in order to promote interoperability among models deployed as services.

Use case scenarios for streamlining the computational drug discovery protocol

Workflows for computational drug discovery

In a real-life scenario, a typical research project in computational drug discovery involves the use of several software, programs and tools that spans from reading input files, data pre-processing, one or more rounds of computation and post-analyses. This would likely involve preprocessing and connecting the outputs of one software or tool as input to another software or tool. Such task may be a troublesome endeavor that may require manual preprocessing of the output and input files. Such issue may potentially be solved if software or tool developers also consider the practical use case scenario pertaining to the interoperability of input/output files for various software and tools.

In cheminformatics research, there are efforts to establish standardized formats and repositories for QSAR models and data. In order to foster reproducible QSAR, exchange formats for data, models, and parameters are needed. QSAR-ML is an XML-based exchange format aimed at promoting interoperable and reproducible QSAR data sets, building on an open and extensible descriptor ontology [246]. The QSAR DataBank (QsarDB) [247, 248] is a repository that aims towards making QSAR modelling transparent, reproducible and accessible via a custom file format and services.The QSAR Model Reporting Format (QMRF) is a harmonised template for summarising and reporting key information on QSAR models, including the results of any validation studies. The information is structured according to the OECD validation principles and is used by the JRC QSAR Model Database [249]. QMRF version 3.0.0 has been updated within the context of the eNanoMapper project [250].

There are also additional general exchange formats for machine learning that are relevant for predictive models in cheminformatics. Predictive Model Markup Language (PMML) [251] is an XML-based predictive model interchange format that also includes data transformations (pre- and post-processing). PMML is sustained by the Data Mining Group [252]. The latest version of QMRF has basic support for PMML. The KNIME workflow software also has support for PMML [253] and the QSAR DataBank (QsarDB) [247, 248] also supports the exporting of models in the PMML data format. A more recent format is the Open Neural Network Exchange (ONNX) that provides an open source format for AI models (i.e. both deep learning and traditional machine learning) [254]. So far there is no reported usage within cheminformatics but the increasing interest in deep learning makes this a relevant candidate for future exchange of models.

In regards to QSAR workflows, there have been considerable efforts directed at this important endeavor that typically entails the utilization of several programs and tools and a series of intricate data pre-processing, model building and analyses (Table 2). Stålring et al. [255] presented an open source machine learning application called AZOrange that allows QSAR model building in a graphical programming environment. Dixon et al. [256] proposed the AutoQSAR as an automated machine learning tool for QSAR modeling using best practice guidelines that was validated on six biological end-points. Nantasenamat et al. [257] reported the development of an automated data mining software for QSAR modeling called AutoWeka that is based on the machine learning software Weka [258]. Kausar and Falcao [259] presents an automated framework based on KNIME for QSAR modeling entailing data pre-processing, model building and validation. Dong et al. [260] introduced an online platform for QSAR modeling known as ChemSAR that is capable of handling chemical structures, computing molecular descriptors, model building as well as producing result plots. Tsiliki et al. [261] proposed an R package known as RRegrs for building multiple regression models using a pre-configured and customizable workflow. Murrell et al. [262] introduced an R package known as the Chemically Aware Model Builder (camb) that continues where the general-purpose R package RRegrs left off which is the capacity to handle chemical structures (i.e. desalting and tautomerizing chemical structures as well as computing molecular descriptors).

Table 2 List of software and packages that implements an automated QSAR modeling workflow

Software/tool	Description	URL	Refs.
Standalone and online applications			
AZOrange	Graphical programming environment based on the Python package "Orange" for performing QSAR modeling workflow	https://github.com/AZcompTox/AZOrange/	[255]
AutoQSAR	Automated machine learning tool for QSAR modeling using best practice guidelines	https://www.schrodinger.com/autoqsar/	[256]
AutoWeka	Automated data mining software for QSAR modeling based on the machine learning software Weka	https://www.mt.mahidol.ac.th/autoweka/	[257]
ChemSAR	Online platform for QSAR modeling that is capable of handling chemical structures, computing molecular descriptors, model building as well as producing result plots	http://chemsar.scbdd.com/	[260]
Tools implemented in R language			
camb	R package that is capable of handling chemical structures, compute descriptors and build QSAR models	https://github.com/cambDI/camb/	[262]
Ezqsar	R package for building QSAR models	https://github.com/enanomapper/RRegrs/	[263]
RRegrs	R package for building multiple regression models using pre-configured and customizable workflow	https://github.com/enanomapper/RRegrs/	[261]

Shamsara [263] presents yet another R package for QSAR modeling called Ezqsar.

Additionally, easy to follow/share pipelines for drug discovery is largely facilitated by the open source nature of the above mentioned cheminformatics and structural biology workflows. Recently, one of us published a book chapter on the construction of reproducible QSAR models [264] in which key factors influencing the reproducibility of QSAR models (i.e. data set, chemical representation, descriptors used, model's parameters/details, predicted endpoint values and data splits) and guidelines on using Jupyter notebook for building reproducible QSAR models are provided. As such, Jupyter notebook is a popular platform in which these workflows are coded, owing to its intuitive blend of code and documentation. Particularly, the ten simple rules for best practice in documenting cheminformatics research using the Jupyter notebook is a useful and timely guideline [265]. These documentations can also be found on GitHub, where a number of researchers share the code to their project's workflow. A selected group of such researchers and the specific area of computational drug discovery research (e.g. ligand-, structure- and/or systems-based) are summarized in Table 3. From this table, we can see that Greg Landrum [266] has shared Jupyter notebooks pertaining to the use of the RDKit module [267] in the context of ligand-based drug discovery on his personal GitHub as well as contributing to the RDKit GitHub [268]). In addition, the OpenEye Python Cookbook [269] is a collection of practical solutions to ligand- and structure-based drug discovery research (i.e. combinatorial library generation, substructure search as well as ligand and protein-ligand structure visualization). Furthermore, myChEMBL [230] is an open source virtual machine that

Table 3 List of selected GitHub URLs of researchers working in the domain of computational drug discovery

Researcher's name	GitHub URL	Ligand-based	Structure-based	Systems-based
Andrea Volkamer	https://github.com/volkamerlab/	✓	✓	
Chanin Nantasenamat	https://github.com/chaninlab/	✓		✓
	https://github.com/chaninn/	✓		
Egon Willighagen	https://github.com/egonw/	✓		
George Papadatos	https://github.com/madgpap/	✓		
Greg Landrum	https://github.com/greglandrum/	✓		
Jan H. Jansen	https://github.com/jensengroup/	✓	✓	
John Chodera	https://github.com/choderalab/	✓	✓	
Ola Spjuth	https://github.com/olas/	✓		
Rajarshi Guha	https://github.com/rajarshi/	✓		
Samo Turk	https://github.com/samoturk/	✓		

combines bioactivity data from ChEMBL with the latest RDKit [267] cheminformatics libraries to sustain a self-contained and user-friendly interface. Putting a new twist to conventional Jupyter notebook, Squonk [270] is a web-based workflow tool based on Jupyter notebook for computational chemistry and cheminformatics for processes encompassing ligand- (i.e. combinatorial library generation, 3D conformer generation, prediction of metabolism and toxicology, molecular property prediction, data visualization and analysis as well as clustering and diversity analysis) and structure-based virtual screening (i.e. scoring active site conformation of compounds).

Aside from the research aspect, educational code-based tutorials on computational drug discovery has been initiated using the Java-based Chemistry Development Kit (CDK) [204, 205, 271] as implemented by the Teach-Discover-Treat (TDT) initiative [272]. This resulted in the development of Python-based tutorials pertaining to the virtual screening workflow to identify malarial drugs [273, 274]. Furthermore, the recently launched TeachOpenCADD platform [275] complements the already available resources by providing students and researchers who are new to computational drug discovery and/or programming with step-by-step *talktorials* that cover both ligand- and structure-based approaches using Python-based open source packages in interactive Jupyter notebooks [276].

Similarly, a software platform in structural bioinformatics known as Biskit [277] links several common tasks in molecular simulation (i.e. each task is a modular object) into a complex workflow that allows streamlined execution of these tasks in a concerted manner. Particularly, researchers can pre-process and analyze macromolecular structures, protein complexes and molecular dynamics trajectories via automated workflow making use of established programs like Xplor, Amber, Hex, DSSP, Fold-X, T-Coffee, TMAlign and Modeller.

In summary, the use of these computational workflows (i.e. that have been tailored to rigorously handle the specific task of interest such as building QSAR models, pre-processing protein structures for molecular simulations, etc.) further helps to ensure the computational reproducibility of the procedures as they have been pre-configured to do so.

Web servers for computational drug discovery

In recent years, the advent of web technologies and the convenience with which users can make use of the functionalities of web-based applications has led to the development of a wide range of web tools and applications in the realm of bioinformatics and cheminformatics for aiding drug discovery efforts (Table 4). The obvious advantage of these web applications is that there is no hassle for installing and maintaining their own computational infrastructure for performing such tasks. The extent of these tools can fall into any one or more of the following tasks: data curation, pre-processing, prediction and analysis. Moreover, another advantage borne from this is the fact that such web applications support reproducibility in that the underlying protocol being performed by the

Table 4 List of selected web applications for handling various bioinformatic and cheminformatic tasks belonging to either ligand-based or structure-based drug design approach

Web servers	Description	URL	Refs.
Ligand-based drug design			
BioTriangle	Compute descriptors for compounds, protein, DNA and their interaction cross-terms	http://biotriangle.scbdd.com/	[285]
ChemDes	Computes 3679 molecular descriptors and 59 fingerprint types for compounds	http://www.scbdd.com/chemdes/	[286]
ChemBench	Enables QSAR model building via pre-defined workflow	http://chembench.mml.unc.edu/	[287]
OCHEM	Online platform providing storage for QSAR data and workflow for model building	http://www.ochem.eu/	[288]
PUMA	Performs analysis and visualization of chemical diversity	https://www.difacquim.com/d-tools/	[289]
Structure-based drug design			
HADDOCK	Performs information-driven docking of biomolecular complexes (e.g. DNA, proteins, peptides, etc.)	http://haddock.science.uu.nl/services/HADDOCK2.2/	[290]
FlexServ	Performs coarse-grained determination of protein dynamics	http://mmb.pcb.ub.es/FlexServ/	[291]
MDWeb	Provides standard protocol for preparing structures, run standard molecular dynamics simulations and analyze trajectories	http://mmb.irbbarcelona.org/MDWeb/	[292]
PoseView	Displays simple molecular interaction diagram of protein-ligand complexes	http://www.zbh.uni-hamburg.de/poseview	[293]
SwissModel	Predicts protein structures via template-based homology	https://swissmodel.expasy.org/	[294]

tool is iteratively executed in the same manner regardless of the number of times it is initiated. In efforts to facilitate easier dissemination of bioinformatic applications as web server, Daniluk et al. [278] introduced the WeBIAS platform, which is a self-contained solution that helps to make command-line programs accessible via web forms. In spite of its advantages and potential utility for the scientific community, the only downside of web databases and applications is the possibility that they may be discontinued at any time. In fact, a recent review explores this issue in which Ősz et al. [279] investigated 3649 web-based services published between 1994 and 2017 and discovered that one-third of these web-based services went out of service. Such discontinued support of web tools and resources poses a great impediment to research reproducibility.

In recent years, the availability of Shiny [280] and Dash [281] packages for the R and Python programming environment, respectively, has greatly lowered the technical barrier to web development for typical R and Python users by facilitating the rapid prototyping of computational workflows as a sharable web-based application. Plotly [282] represents a robust tool for producing interactive data visualization that can be collaboratively shared to colleagues. Graphs and dashboards can be made with no coding and is thus appealing to the non-technical users while the available Plotly packages for various platforms (e.g. R, Python, Javascript and React) is equally appealing to technical users as well.

Conclusion

The dawn of the big data era in drug discovery is made possible by technological advancements in the various omics disciplines. Such big data brings with it great opportunities for advancing life sciences while at the same time bringing several potential problems pertaining to the reliability and reproducibility of generated results. In efforts to steer clear of the potential pitfalls that may be lurking ahead, it is of great importance to grasp the current state-of-the-art of research reproducibility in computational drug discovery as to ensure that the underlying work is of high quality and that it is capable of withstanding reproduction of the described methodology by external research group. A wide range of resources and tools are available for embarking on the journey towards reproducibility in computational drug discovery projects, which has been explored in this review article. The growing culture of sharing the underlying data and codes published in research articles pertaining to computational drug discovery is anticipated to drive the field forward as new and useful knowledge base can gradually be built on top of its predecessors thereby creating a snowball effect. In recent years, policies imposed by granting agencies and publishers are in favor of data and code sharing, which are further facilitated by third-party platforms (e.g. Authorea, Code Ocean, Jupyter notebook, Manuscripts.io, etc.) that further enhances reproducibility in which manuscripts and codes that are shared on the web are no longer static files waiting to be downloaded but are *"living"* codes and documents that can dynamically be edited and executed in real-time.

In summary, we have attempted to detail the diverse range of issues faced by the predictive modelling community in its role to develop and deploy efficient and reliable computational tools for drug discovery. From examples presented herein, it is clear that close interaction between frontline drug discovery scientists, the intermediate data modellers, and back office computer scientists and administrators. The challenge that each of these groups faces are quite different in nature and thus there needs to be improved understanding of these issues and a common vocabulary in order to maximize their impact. This is no small task, given the breadth of the fields involved. We note that it is of critical importance that data modellers, tool developers and administrators do not lose sight of the fact that tools must be developed for use by front line scientists in day-to-day, dynamic environment. This dynamic nature may lead to a degree of conflict with best practices espoused by the data science community (i.e. due to ever changing needs).

With this in mind, it is necessary to understand that certain solutions are preferable to the developer community and may not be considered optimal to model developers. For example, custom models using user-derived descriptors (i.e. experimental data or non-standard 3D computational models) may be desirable, but difficult to incorporate rapidly into QSAR models in a short period of time. Alternatively, predictive models that deliver lower overall predictive performance, but greater interpretability, may be preferred in some cases. The latter model types might not appear in automated solutions in now common modelling workflows as selection conditions are generally driven by statistical considerations rather than needs of the end user.

Open source promotes transparency in implementations and allows for easy access to validate analysis. When working with data and modeling, it is often difficult to keep track of tools and parameters used in the analysis. Workflow systems can aid in this and are gaining momentum in drug discovery. They contribute to more robust multi-step computations, transparency, provenance and ease of reproducibility. There is also an increased push for interoperability and standardization of workflow specifications with projects like Common Workflow Language.

With growing data sizes, the use of shared or public computing infrastructures (HPC/Cloud) is necessary and therefore adds another level of complexity for computational reproducibility. In order for all tools used for data analysis to be portable between systems, technologies such as virtual machines and software containers are widely used. When connecting containers and virtual machines with workflow systems, a high level of automation can be achieved, and through that improved reproducibility. Virtual infrastructure and containers also facilitate more reliable and replicable services, for instance, for deploying models as services over the network.

Acknowledgements
This work is supported by the Research Career Development Grant (No. RSA6280075) from the Thailand Research Fund. The authors would also like to thank Dr. Sirarat Sarntivijai from the European Bioinformatics Institute and Dr. Likit Preeyanon from the Department of Community Medical Technology for fruitful discussions.

Authors' contributions
CN conceived the study. OS, MPG and CN conceptualized the study. All authors reviewed the literature and drafted the manuscript. SL, SS and CN prepared the figures. CN vetted the manuscript. All authors read and approved the final manuscript.

Availability of data and materials
Not applicable.

Competing interests
The authors declare that they have no competing interests.

Author details
[1] Center of Data Mining and Biomedical Informatics, Faculty of Medical Technology, Mahidol University, 10700 Bangkok, Thailand. [2] Department of Pharmaceutical Biosciences, Uppsala University, 751 24 Uppsala, Sweden. [3] Interdisciplinary Graduate Program in Bioscience, Faculty of Science, Kasetsart University, 10900 Bangkok, Thailand. [4] Department of Biomedical Engineering, Faculty of Engineering, King Mongkut's Institute of Technology Ladkrabang, 10520 Bangkok, Thailand.

Received: 17 July 2019 Accepted: 2 January 2020
Published online: 28 January 2020

References

1. Mullard A (2016) Biotech R&D spend jumps by more than 15. Nat Rev Drug Discov 15(7):447. https://doi.org/10.1038/nrd.2016.135
2. Stratmann HG (2010) Bad medicine: when medical research goes wrong. Analog Sci Fict Fact CXXX(9):20–30
3. DiMasi JA, Grabowski HG, Hansen RW (2016) Innovation in the pharmaceutical industry: new estimates of R&D costs. J Health Econ 47:20–33. https://doi.org/10.1016/j.jhealeco.2016.01.012
4. Biotechnology Innovation Organisation (2016) Clinical Development Success Rates 2006–2015
5. Ogu CC, Maxa JL (2000) Drug interactions due to cytochrome p450. Baylor Univ Med Center Proc 13(4):421–423. https://doi.org/10.1080/08998280.2000.11927719
6. Fox S, Farr-Jones S, Sopchak L, Boggs A, Nicely HW, Khoury R, Biros M (2006) High-throughput screening: update on practices and success. J Biomol Screen 11(7):864–869. https://doi.org/10.1177/1087057106292473
7. Hughes JP, Rees S, Kalindjian SB, Philpott KL (2011) Principles of early drug discovery. Br J Pharmacol 162(6):1239–1249. https://doi.org/10.1111/j.1476-5381.2010.01127.x
8. Ruddigkeit L, van Deursen R, Blum LC, Reymond J-L (2012) Enumeration of 166 billion organic small molecules in the chemical universe database gdb-17. J Chem Inform Model 52(11):2864–2875. https://doi.org/10.1021/ci300415d
9. Villoutreix BO, Renault N, Lagorce D, Sperandio O, Montes M, Miteva MA (2007) Free resources to assist structure-based virtual ligand screening experiments. Curr Protein Pept Sci 8(4):381–411
10. Nantasenamat C, Prachayasittikul V (2015) Maximizing computational tools for successful drug discovery. Expert Opin Drug Discov 10(4):321–329. https://doi.org/10.1517/17460441.2015.1016497
11. Feng BY, Simeonov A, Jadhav A, Babaoglu K, Inglese J, Shoichet BK, Austin CP (2007) A high-throughput screen for aggregation-based inhibition in a large compound library. J Med Chem 50(10):2385–2390. https://doi.org/10.1021/jm061317y
12. Soares KM, Blackmon N, Shun TY, Shinde SN, Takyi HK, Wipf P, Lazo JS, Johnston PA (2010) Profiling the nih small molecule repository for compounds that generate H_2O_2 by redox cycling in reducing environments. Assay Drug Dev Technol 8(2):152–174. https://doi.org/10.1089/adt.2009.0247
13. Young D, Martin T, Venkatapathy R, Harten P (2008) Are the chemical structures in your QSAR correct? QSAR Combinatorial Sci 27(11–12):1337–1345. https://doi.org/10.1002/qsar.200810084
14. Zhao L, Wang W, Sedykh A, Zhu H (2017) Experimental errors in QSAR modeling sets: what we can do and what we cannot do. ACS Omega 2(6):2805–2812. https://doi.org/10.1021/acsomega.7b00274
15. Clark RD (2019) A path to next-generation reproducibility in chem-informatics. J Cheminform 11:62. https://doi.org/10.1186/s13321-019-0385-0
16. Walters P (2019) Where's the code? http://practicalcheminformatics.blogspot.com/2019/05/wheres-code.html. Accessed 1 Nov 2019
17. Garabedian TE (1997) Laboratory record keeping. Nat Biotechnol 15(8):799–800. https://doi.org/10.1038/nbt0897-799
18. Plavén-Sigray P, Matheson GJ, Schiffler BC, Thompson WH (2017) The readability of scientific texts is decreasing over time. eLife. https://doi.org/10.7554/eLife.27725
19. Dirnagl U, Przesdzing I (2016) A pocket guide to electronic laboratory notebooks in the academic life sciences. F1000 Res 5:2 https://doi.org/10.12688/f1000research.7628.1
20. Rubacha M, Rattan AK, Hosselet SC (2011) A review of electronic laboratory notebooks available in the market today. J Lab Autom 16(1):90–98. https://doi.org/10.1016/j.jala.2009.01.002
21. Mascarelli A (2014) Research tools: jump off the page. Nature 507(7493):523–525. https://doi.org/10.1038/nj7493-523a
22. Schnell S (2015) Ten simple rules for a computational biologist's laboratory notebook. PLoS Comput Biol 11(9):1004385. https://doi.org/10.1371/journal.pcbi.1004385
23. Bradley J-C, Neylon C (2008) Data on display. Interview by Katherine Sanderson. Nature 455(7211):273. https://doi.org/10.1038/455273a
24. Butler D (2005) Electronic notebooks: a new leaf. Nature 436(7047):20–21. https://doi.org/10.1038/436020a
25. Project Jupyter (2019) The Jupyter Notebook. http://www.jupyter.org/. Accessed 9 Jan 2019
26. Project Jupyter (2019) nbviewer. http://nbviewer.jupyter.org/. Accessed 9 Jan 2019
27. Freeman Lab (2019) Binder. http://mybinder.org/. Accessed 9 Jan 2019
28. Google (2019) Colaboratory. https://colab.research.google.com/. Accessed 9 Jan 2019
29. Baker M (2016) 1,500 scientists lift the lid on reproducibility. Nature 533(7604):452–454. https://doi.org/10.1038/533452a
30. Head ML, Holman L, Lanfear R, Kahn AT, Jennions MD (2015) The extent and consequences of p-hacking in science. PLoS Biol 13(3):1002106. https://doi.org/10.1371/journal.pbio.1002106
31. Simonsohn U, Nelson LD, Simmons JP (2014) P-curve: a key to the file-drawer. J Exp Psychol Gen 143(2):534–547. https://doi.org/10.1037/a0033242
32. Ioannidis JPA (2008) Effect of formal statistical significance on the credibility of observational associations. Am J Epidemiol 168(4):374–83384. https://doi.org/10.1093/aje/kwn156

33. Risch NJ (2000) Searching for genetic determinants in the new millennium. Nature 405(6788):847–856. https://doi.org/10.1038/35015718
34. Wacholder S, Chanock S, Garcia-Closas M, El Ghormli L, Rothman N (2004) Assessing the probability that a positive report is false: an approach for molecular epidemiology studies. J Natl Cancer Inst 96(6):434–442
35. Wilkinson MD, Dumontier M, Aalbersberg IJ, Appleton G, Axton M, Baak A, Blomberg N, Boiten JW, da Silva Santos LB, Bourne PE, Bouwman J, Brookes AJ, Clark T, Crosas M, Dillo I, Dumon O, Edmunds S, Evelo CT, Finkers R, Gonzalez-Beltran A, Gray AJ, Groth P, Goble C, Grethe JS, Heringa J, 't Hoen PA, Hooft R, Kuhn T, Kok R, Kok J, Lusher SJ, Martone ME, Mons A, Packer AL, Persson B, Rocca-Serra P, Roos M, van Schaik R, Sansone SA, Schultes E, Sengstag T, Slater T, Strawn G, Swertz MA, Thompson M, van der Lei J, van Mulligen E, Velterop J, Waagmeester A, Wittenburg P, Wolstencroft K, Zhao J, Mons B (2016) The FAIR Guiding Principles for scientific data management and stewardship. Sci Data 3:160018. https://doi.org/10.1038/sdata.2016.18
36. Guha R, Willighagen E (2017) Helping to improve the practice of cheminformatics. J Cheminform 9(1):40. https://doi.org/10.1186/s13321-017-0217-z
37. Collin's English Dictionary (2019) Reproduce. http://www.dictionary.com/browse/reproducibility. Accessed 9 Jan 2019
38. Schwab M, Karrenbach M, Claerbout J (2000) Making scientific computations reproducible. Comput Sci Eng 2:61–67
39. Casadevall A, Fang FC (2010) Reproducible science. Infect Immun 78(12):4972–4975. https://doi.org/10.1128/IAI.00908-10
40. Kerr Bernal S (2006) A massive snowball of fraud and deceit. J Androl 27(3):313–315. https://doi.org/10.2164/jandrol.06007
41. Joint Committee for Guides in Metrology (2008) Evaluation of measurement data — Guide to the expression of uncertainty in measurement. https://www.bipm.org/utils/common/documents/jcgm/JCGM_100_2008_E.pdf. Accessed 1 Nov 2019
42. Oudeyer P-Y, Merrick K (2016) Computational modelling across disciplines. IEEE Cogn Dev Syst Newslett 13(2):1
43. Taylor CF, Field D, Sansone SA, Aerts J, Apweiler R, Ashburner M, Ball CA, Binz PA, Bogue M, Booth T, Brazma A, Brinkman RR, Michael Clark A, Deutsch EW, Fiehn O, Fostel J, Ghazal P, Gibson F, Gray T, Grimes G, Hancock JM, Hardy NW, Hermjakob H, Julian RK, Kane M, Kettner C, Kinsinger C, Kolker E, Kuiper M, Le Novere N, Leebens-Mack J, Lewis SE, Lord P, Mallon AM, Marthandan N, Masuya H, McNally R, Mehrle A, Morrison N, Orchard S, Quackenbush J, Reecy JM, Robertson DG, Rocca-Serra P, Rodriguez H, Rosenfelder H, Santoyo-Lopez J, Scheuermann RH, Schober D, Smith B, Snape J, Stoeckert CJ, Tipton K, Sterk P, Untergasser A, Vandesompele J, Wiemann S (2008) Promoting coherent minimum reporting guidelines for biological and biomedical investigations: the MIBBI project. Nat Biotechnol 26(8):889–896. https://doi.org/10.1038/nbt.1411
44. Hermjakob H, Montecchi-Palazzi L, Bader G, Wojcik J, Salwinski L, Ceol A, Moore S, Orchard S, Sarkans U, von Mering C, Roechert B, Poux S, Jung E, Mersch H, Kersey P, Lappe M, Li Y, Zeng R, Rana D, Nikolski M, Husi H, Brun C, Shanker K, Grant SG, Sander C, Bork P, Zhu W, Pandey A, Brazma A, Jacq B, Vidal M, Sherman D, Legrain P, Cesareni G, Xenarios I, Eisenberg D, Steipe B, Hogue C, Apweiler R (2004) The HUPO PSI's molecular interaction format—a community standard for the representation of protein interaction data. Nat Biotechnol 22(2):177–183. https://doi.org/10.1038/nbt926
45. Demir E, Cary MP, Paley S, Fukuda K, Lemer C, Vastrik I, Wu G, D'Eustachio P, Schaefer C, Luciano J, Schacherer F, Martinez-Flores I, Hu Z, Jimenez-Jacinto V, Joshi-Tope G, Kandasamy K, Lopez-Fuentes AC, Mi H, Pichler E, Rodchenkov I, Splendiani A, Tkachev S, Zucker J, Gopinath G, Rajasimha H, Ramakrishnan R, Shah I, Syed M, Anwar N, Babur O, Blinov M, Brauner E, Corwin D, Donaldson S, Gibbons F, Goldberg R, Hornbeck P, Luna A, Murray-Rust P, Neumann E, Ruebenacker O, Reubenacker O, Samwald M, van Iersel M, Wimalaratne S, Allen K, Braun B, Whirl-Carrillo M, Cheung KH, Dahlquist K, Finney A, Gillespie M, Glass E, Gong L, Haw R, Honig M, Hubaut O, Kane D, Krupa S, Kutmon M, Leonard J, Marks D, Merberg D, Petri V, Pico A, Ravenscroft D, Ren L, Shah N, Sunshine M, Tang R, Whaley R, Letovksy S, Buetow KH, Rzhetsky A, Schachter V, Sobral BS, Dogrusoz U, McWeeney S, Aladjem M, Birney E, Collado-Vides J, Goto S, Hucka M, Le Novere N, Maltsev N, Pandey A, Thomas P, Wingender E, Karp PD, Sander C, Bader GD (2010) The BioPAX community standard for pathway data sharing. Nat Biotechnol 28(9):935–942. https://doi.org/10.1038/nbt.1666
46. Wf4Ever Project (2019) Wf4Ever github repository. http://wf4ever.github.io/. Accessed 9 Jan 2019
47. Cooper J, Vik JO, Waltemath D (2015) A call for virtual experiments: accelerating the scientific process. Progr Biophys Mol Biol 117(1):99–106. https://doi.org/10.1016/j.pbiomolbio.2014.10.001
48. Gentleman RC, Carey VJ, Bates DM, Bolstad B, Dettling M, Dudoit S, Ellis B, Gautier L, Ge Y, Gentry J et al (2004) Bioconductor: open software development for computational biology and bioinformatics. Genome Biol 5(10):80. https://doi.org/10.1186/gb-2004-5-10-r80
49. Blankenberg D, Von Kuster G, Coraor N, Ananda G, Lazarus R, Mangan M, Nekrutenko A, Taylor J (2010) Galaxy: a web-based genome analysis tool for experimentalists. Curr Protoc Mol Biol Chapt 19:19–10121. https://doi.org/10.1002/0471142727.mb1910s89
50. Giardine B, Riemer C, Hardison RC, Burhans R, Elnitski L, Shah P, Zhang Y, Blankenberg D, Albert I, Taylor J et al (2005) Galaxy: a platform for interactive large-scale genome analysis. Genome Res 15(10):1451–1455. https://doi.org/10.1101/gr.4086505
51. Goecks J, Nekrutenko A, Taylor J (2010) Galaxy Team: Galaxy: a comprehensive approach for supporting accessible, reproducible, and transparent computational research in the life sciences. Genome Biol 11(8):86. https://doi.org/10.1186/gb-2010-11-8-r86
52. Bouckaert R, Heled J, Kühnert D, Vaughan T, Wu C-H, Xie D, Suchard MA, Rambaut A, Drummond AJ (2014) Beast 2: a software platform for bayesian evolutionary analysis. PLoS Comput Biol 10(4):1003537. https://doi.org/10.1371/journal.pcbi.1003537
53. Bouckaert R, Vaughan TG, Barido-Sottani J, Duchêne S, Fourment M, Gavryushkina A, Heled J, Jones G, Kühnert D, De Maio N, Matschiner M, Mendes FK, Müller NF, Ogilvie HA, du Plessis L, Popinga A, Rambaut A, Rasmussen D, Siveroni I, Suchard MA, Wu C-H, Xie D, Zhang C, Stadler T, Drummond AJ (2019) Beast 2.5: An advanced software platform for bayesian evolutionary analysis. PLoS Comput Biol 15(4):1006650. https://doi.org/10.1371/journal.pcbi.1006650
54. Teytelman L protocols.io - the #1 science methods repository
55. High Level Expert Group on Scientific Data (2010) Riding the Wave—how Europe can gain from the rising tide of scientific data. https://www.fosteropenscience.eu/content/riding-wave-how-europe-can-gain-rising-tide-scientific-data/. Accessed 9 Jan 2019
56. National Institutes of Health (2019) NIH Grants Policy Statement. https://grants.nih.gov/policy/nihgps/index.htm. Accessed 9 Jan 2019
57. NordForsk (2019) Open Access to Research Data - Status, Issues and Outlook. https://www.nordforsk.org/en/publications/publications_container/open-access-to-research-data-2013-status-issues-and-outlook/. Accessed 9 Jan 2019
58. Borgman CL (2015) Big data, little data, no data: scholarship in the networked world. MIT Press, Cambridge
59. Margolis R, Derr L, Dunn M, Huerta M, Larkin J, Sheehan J, Guyer M, Green ED (2014) The national institutes of health's big data to knowledge (bd2k) initiative: capitalizing on biomedical big data. J Am Med Inform Assoc 21(6):957–958. https://doi.org/10.1136/amiajnl-2014-002974
60. Pasquetto IV, Randles BM, Borgman CL (2017) On the reuse of scientific data. Data Sci J. https://doi.org/10.5334/dsj-2017-008
61. Wallis JC, Rolando E, Borgman CL (2013) If we share data, will anyone use them? data sharing and reuse in the long tail of science and technology. PLoS ONE 8(7):67332. https://doi.org/10.1371/journal.pone.0067332
62. Chavan V, Penev L (2011) The data paper: a mechanism to incentivize data publishing in biodiversity science. BMC Bioinform 12 Suppl 15:2. https://doi.org/10.1186/1471-2105-12-S15-S2
63. Gorgolewski KJ, Margulies DS, Milham MP (2013) Making data sharing count: a publication-based solution. Front Neurosci 7:9. https://doi.org/10.3389/fnins.2013.00009
64. Searls DB (2010) The roots of bioinformatics. PLoS Comput Biol 6(6):1000809. https://doi.org/10.1371/journal.pcbi.1000809
65. Kanwal S, Khan FZ, Lonie A, Sinnott RO (2017) Investigating reproducibility and tracking provenance—a genomic workflow case study. BMC Bioinform 18(1):337. https://doi.org/10.1186/s12859-017-1747-0
66. Kim Y-M, Poline J-B, Dumas G (2017) Experimenting with reproducibility in bioinformatics. BioRxiv. https://doi.org/10.1101/143503

67. Sandve GK, Nekrutenko A, Taylor J, Hovig E (2013) Ten simple rules for reproducible computational research. PLoS Comput Biol 9(10):1003285. https://doi.org/10.1371/journal.pcbi.1003285
68. Van Neste C, Gansemans Y, De Coninck D, Van Hoofstat D, Van Criekinge W, Deforce D, Van Nieuwerburgh F (2015) Forensic massively parallel sequencing data analysis tool: implementation of MyFLq as a standalone web- and Illumina BaseSpace®-application. Forensic Sci Int Genet 15:2–7. https://doi.org/10.1016/j.fsigen.2014.10.006
69. Dove ES, Joly Y, Tassé A-M (2015) Public Population Project in Genomics and Society (P3G) International Steering Committee and International Cancer Genome Consortium (ICGC) Ethics and Policy Committee, Knoppers, B.M.: genomic cloud computing: legal and ethical points to consider. Eur J Human Genet 23(10):1271–1278. https://doi.org/10.1038/ejhg.2014.196
70. Docker Inc. (2019) Docker. https://www.docker.com/. Accessed 9 Jan 2019
71. da Veiga Leprevost F, Gruning BA, Alves Aflitos S, Rost HL, Uszkoreit J, Barsnes H, Vaudel M, Moreno P, Gatto L, Weber J, Bai M, Jimenez RC, Sachsenberg T, Pfeuffer J, Vera Alvarez R, Griss J, Nesvizhskii AI, Perez-Riverol Y (2017) BioContainers: an open-source and community-driven framework for software standardization. Bioinformatics 33(16):2580–2582. https://doi.org/10.1093/bioinformatics/btx192
72. Kim B, Ali T, Lijeron C, Afgan E, Krampis K (2017) Bio-docklets: virtualization containers for single-step execution of ngs pipelines. GigaScience 6(8):1–7. https://doi.org/10.1093/gigascience/gix048
73. Menegidio FB, Jabes DL, de Oliveira R Costa, Nunes LR (2018) Dugong: a Docker image, based on Ubuntu Linux, focused on reproducibility and replicability for bioinformatics analyses. Bioinformatics 34(3):514–515. https://doi.org/10.1093/bioinformatics/btx554
74. Kulkarni N, Alessandri L, Panero R, Arigoni M, Olivero M, Ferrero G, Cordero F, Beccuti M, Calogero RA (2018) Reproducible bioinformatics project: a community for reproducible bioinformatics analysis pipelines. BMC Bioinform 19(Suppl 10):349. https://doi.org/10.1186/s12859-018-2296-x
75. Rozenblatt-Rosen O, Stubbington MJT, Regev A, Teichmann SA (2017) The Human Cell Atlas: from vision to reality. Nature 550(7677):451–453. https://doi.org/10.1038/550451a
76. Peng RD (2011) Reproducible research in computational science. Science 334(6060):1226–1227. https://doi.org/10.1126/science.1213847
77. Stodden V, Leisch F, Peng RD (2014) Implementing reproducible research. CRC Press/Taylor & Francis Group, Boca Raton
78. Scientific Data (2019) Recommended Data Repositories. https://www.nature.com/sdata/policies/repositories/. Accessed 9 Jan 2019
79. Dryad (2019) Dryad Digital Repository. https://datadryad.org/. Accessed 9 Jan 2019
80. Dryad (2019) DryadLab. http://datadryad.org/pages/dryadlab/. Accessed 9 Jan 2019
81. figshare (2019) figshare—credit for all your research. http://www.figshare.com/. Accessed 9 Jan 2019
82. Singh J (2011) Figshare. J Pharmacol Pharmacother 2(2):138–139. https://doi.org/10.4103/0976-500X.81919
83. Zenodo (2019) Zenodo—Research. Shared. https://zenodo.org/. Accessed 9 Jan 2019
84. Open Science Framework (2019) OSF Home. https://osf.io/. Accessed 9 Jan 2019
85. Center for Open Science (2019) Center for Open Science Website. https://cos.io/. Accessed 9 Jan 2019
86. Foster ED, Deardorff A (2017) Open science framework (osf). J Med Lib Assoc 105(2):203–206. https://doi.org/10.5195/JMLA.2017.88
87. Macmillan Publishers Limited (2019) Scientific Data. https://www.nature.com/sdata/. Accessed 9 Jan 2019
88. Elsevier (2019) Data in Brief. https://www.journals.elsevier.com/data-in-brief/. Accessed 9 Jan 2019
89. MDPI (2019) Data. http://www.mdpi.com/journal/data/. Accessed 9 Jan 2019
90. F1000Research (2019) F1000Research | Open Access Publishing Platform | Beyond a Research Journal. https://f1000research.com/. Accessed 9 Jan 2019
91. arXiv (2019) arXiv.org e-Print archive. https://arxiv.org/. Accessed 9 Jan 2019
92. bioRxiv (2019) bioRxiv.org—the preprint server for Biology. https://www.biorxiv.org/. Accessed 9 Jan 2019
93. ChemRxiv (2019) ChemRxiv: the Preprint Server for Chemistry. https://chemrxiv.org/. Accessed 9 Jan 2019
94. PeerJ (2019) PeerJ Preprints. https://peerj.com/preprints/. Accessed 9 Jan 2019
95. Bitbucket (2019) Bitbucket - The Git solution for professional teams. https://bitbucket.org/. Accessed 9 Jan 2019
96. GitLab (2019) GitLab. https://about.gitlab.com/. Accessed 9 Jan 2019
97. Assembla (2019) Assembla: Secure Git, Secure Software Development in the Cloud. https://www.assembla.com/. Accessed 9 Jan 2019
98. Google (2019) Cloud Source Repositories. https://cloud.google.com/source-repositories/. Accessed 9 Jan 2019
99. Sofroniew NJ, Vlasov YA, Hires SA, Freeman J, Svoboda K (2015) Neural coding in barrel cortex during whisker-guided locomotion. eLife. https://doi.org/10.7554/eLife.12559
100. Li N, Daie K, Svoboda K, Druckmann S (2016) Robust neuronal dynamics in premotor cortex during motor planning. Nature 532(7600):459–464. https://doi.org/10.1038/nature17643
101. Code Ocean (2019) Code Ocean—Professional tools for researchers. https://codeocean.com/. Accessed 9 Jan 2019
102. Cornell Tech (2019) Code Ocean: Tackling Reproducibility and Transparency in Scientific Research. https://tech.cornell.edu/news/code-ocean-tackling-reproducibility-and-transparency-in- scientific-research. Accessed 9 Jan 2019
103. Perkel J (2019) TechBlog: C. Titus Brown: Predicting the paper of the future. http://blogs.nature.com/naturejobs/2017/06/01/techblog-c-titus-brown-predicting-the-paper-of-the-future/. Accessed 9 Jan 2019
104. Software Carpentry (2019) Software Carpentry—Teaching basic lab skills for research computing. https://software-carpentry.org/. Accessed 9 Jan 2019
105. Data Carpentry (2019) Data Carpentry—Building communities teaching universal data literacy. http://www.datacarpentry.org/. Accessed 9 Jan 2019
106. Birney E, Hudson TJ, Green ED, Gunter C, Eddy S, Rogers J, Harris JR, Ehrlich SD, Apweiler R, Austin CP, Berglund L, Bobrow M, Bountra C, Brookes AJ, Cambon-Thomsen A, Carter NP, Chisholm RL, Contreras JL, Cooke RM, Crosby WL, Dewar K, Durbin R, Dyke SO, Ecker JR, El Emam K, Feuk L, Gabriel SB, Gallacher J, Gelbart WM, Granell A, Guarner F, Hubbard T, Jackson SA, Jennings JL, Joly Y, Jones SM, Kaye J, Kennedy KL, Knoppers BM, Kyrpides NC, Lowrance WW, Luo J, MacKay JJ, Martin-Rivera L, McCombie WR, McPherson JD, Miller L, Miller W, Moerman D, Mooser V, Morton CC, Ostell JM, Ouellette BF, Parkhill J, Raina PS, Rawlings C, Scherer SE, Scherer SW, Schofield PN, Sensen CW, Stodden VC, Sussman MR, Tanaka T, Thornton J, Tsunoda T, Valle D, Vuorio EI, Walker NM, Wallace S, Weinstock G, Whitman WB, Worley KC, Wu C, Wu J, Yu J (2009) Prepublication data sharing. Nature 461(7261):168–170. https://doi.org/10.1038/461168a
107. González-Medina M, Naveja JJ, Sánchez-Cruz N, Medina-Franco JL (2017) Open chemoinformatic resources to explore the structure, properties and chemical space of molecules. RSC Adv 7(85):54153–54163. https://doi.org/10.1039/C7RA11831G
108. Hasegawa K, Funatsu K (2014) Data mining of chemogenomics data using bi-modal PLS methods and chemical interpretation for molecular design. Mol Inform 33(11–12):749–756. https://doi.org/10.1002/minf.201400061
109. Mendez D, Gaulton A, Bento AP, Chambers J, De Veij M, Félix E, Magariños MP, Mosquera JF, Mutowo P, Nowotka M, Gordillo-Marañón M, Hunter F, Junco L, Mugumbate G, Rodriguez-Lopez M, Atkinson F, Bosc N, Radoux CJ, Segura-Cabrera A, Hersey A, Leach AR (2019) ChEMBL: towards direct deposition of bioassay data. Nucleic Acids Res 47(D1):930–940. https://doi.org/10.1093/nar/gky1075
110. Kim S, Chen J, Cheng T, Gindulyte A, He J, He S, Li Q, Shoemaker BA, Thiessen PA, Yu B, Zaslavsky L, Zhang J, Bolton EE (2019) PubChem 2019 update: improved access to chemical data. Nucleic Acids Res 47(D1):1102–1109. https://doi.org/10.1093/nar/gky1033
111. Gilson MK, Liu T, Baitaluk M, Nicola G, Hwang L, Chong J (2016) BindingDB in 2015: a public database for medicinal chemistry, computational chemistry and systems pharmacology. Nucleic Acids Res 44(D1):1045–53. https://doi.org/10.1093/nar/gkv1072

112. Gilson MK (2019) BindingDB. https://www.bindingdb.org. Accessed 9 Jan 2019
113. Ursu O, Holmes J, Knockel J, Bologa CG, Yang JJ, Mathias SL, Nelson SJ, Oprea TI (2017) DrugCentral: online drug compendium. Nucleic Acids Res 45(D1):932–939. https://doi.org/10.1093/nar/gkw993
114. Law V, Knox C, Djoumbou Y, Jewison T, Guo AC, Liu Y, Maciejewski A, Arndt D, Wilson M, Neveu V, Tang A, Gabriel G, Ly C, Adamjee S, Dame ZT, Han B, Zhou Y, Wishart DS (2014) DrugBank 4.0: shedding new light on drug metabolism. Nucleic Acids Res 42(Database issue):1091–1097. https://doi.org/10.1093/nar/gkt1068
115. Wishart DS, Feunang YD, Guo AC, Lo EJ, Marcu A, Grant JR, Sajed T, Johnson D, Li C, Sayeeda Z, Assempour N, Iynkkaran I, Liu Y, Maciejewski A, Gale N, Wilson A, Chin L, Cummings R, Le D, Pon A, Knox C, Wilson M (2018) DrugBank 5.0: a major update to the DrugBank database for 2018. Nucleic Acids Res 46(D1):1074–1082. https://doi.org/10.1093/nar/gkx1037
116. Mathias SL, Hines-Kay J, Yang JJ, Zahoransky-Kohalmi G, Bologa CG, Ursu O, Oprea TI (2013) The CARLSBAD database: a confederated database of chemical bioactivities. Database 2013:044. https://doi.org/10.1093/database/bat044
117. Placzek S, Schomburg I, Chang A, Jeske L, Ulbrich M, Tillack J, Schomburg D (2017) Brenda in 2017: new perspectives and new tools in brenda. Nucleic Acids Res 45(D1):380–388. https://doi.org/10.1093/nar/gkw952
118. Sun J, Jeliazkova N, Chupakin V, Golib-Dzib J-F, Engkvist O, Carlsson L, Wegner J, Ceulemans H, Georgiev I, Jeliazkov V, Kochev N, Ashby TJ, Chen H (2017) ExCAPE-DB: an integrated large scale dataset facilitating big data analysis in chemogenomics. J Cheminform 9:17. https://doi.org/10.1186/s13321-017-0203-5
119. Güner OF (2002) History and evolution of the pharmacophore concept in computer-aided drug design. Curr Top Med Chem 2(12):1321–1332. https://doi.org/10.2174/1568026023392940
120. Patel Y, Gillet VJ, Bravi G, Leach AR (2002) A comparison of the pharmacophore identification programs: catalyst, disco and gasp. J Comput Aided Mol Des 16(8–9):653–681. https://doi.org/10.1023/a:1021954728347
121. Sliwoski G, Kothiwale S, Meiler J, Lowe EW (2014) Computational methods in drug discovery. Pharmacol Rev 66(1):334–395. https://doi.org/10.1124/pr.112.007336
122. Kolossov E, Lemon A (2006) Medicinal chemistry tools: making sense of hts data. Eur J Med Chem 41(2):166–175. https://doi.org/10.1016/j.ejmech.2005.10.005
123. Doke SK, Dhawale SC (2015) Alternatives to animal testing: a review. Saudi Pharm J 23(3):223–229. https://doi.org/10.1016/j.jsps.2013.11.002
124. Cronin MT, Jaworska JS, Walker JD, Comber MH, Watts CD, Worth AP (2003) Use of QSARs in international decision-making frameworks to predict health effects of chemical substances. Environ Health Perspect 111(10):1391–1401. https://doi.org/10.1289/ehp.5760
125. Hofer T, Gerner I, Gundert-Remy U, Liebsch M, Schulte A, Spielmann H, Vogel R, Wettig K (2004) Animal testing and alternative approaches for the human health risk assessment under the proposed new European chemicals regulation. Arch Toxicol 78(10):549–564. https://doi.org/10.1007/s00204-004-0577-9
126. Ashby J (1985) Fundamental structural alerts to potential carcinogenicity or noncarcinogenicity. Environ Mutagen 7(6):919–921. https://doi.org/10.1002/em.2860070613
127. Ashby J, Tennant RW (1991) Definitive relationships among chemical structure, carcinogenicity and mutagenicity for 301 chemicals tested by the U.S. NTP. Mutation Res 257(3):229–306. https://doi.org/10.1016/0165-1110(91)90003-e
128. Devillers J, Mombelli E, Samsera R (2011) Structural alerts for estimating the carcinogenicity of pesticides and biocides. SAR QSAR Environ Res 22(1–2):89–106. https://doi.org/10.1080/1062936X.2010.548349
129. Aptula AO, Patlewicz G, Roberts DW (2005) Skin sensitization: reaction mechanistic applicability domains for structure-activity relationships. Chem Res Toxicol 18(9):1420–1426. https://doi.org/10.1021/tx050075m
130. Roberts DW, Patlewicz G, Kern PS, Gerberick F, Kimber I, Dearman RJ, Ryan CA, Basketter DA, Aptula AO (2007) Mechanistic applicability domain classification of a local lymph node assay dataset for skin sensitization. Chem Res Toxicol 20(7):1019–1030. https://doi.org/10.1021/tx700024w
131. Blake JF (2005) Identification and evaluation of molecular properties related to preclinical optimization and clinical fate. Med Chem 1(6):649–655. https://doi.org/10.2174/157340605774598081
132. Hann M, Hudson B, Lewell X, Lifely R, Miller L, Ramsden N (1999) Strategic pooling of compounds for high-throughput screening. J Chem Inform Comput Sci 39(5):897–902. https://doi.org/10.1021/ci990423o
133. Pearce BC, Sofia MJ, Good AC, Drexler DM, Stock DA (2006) An empirical process for the design of high-throughput screening deck filters. J Chem Inform Model 46(3):1060–1068. https://doi.org/10.1021/ci050504m
134. Alves V, Muratov E, Capuzzi S, Politi R, Low Y, Braga R, Zakharov AV, Sedykh A, Mokshyna E, Farag S, Andrade CH, Kuz'min VE, Fourches D, Tropsha A (2016) Alarms about structural alerts. Green Chem 18(16):4348–4360. https://doi.org/10.1039/C6GC01492E
135. Labute P (2000) A widely applicable set of descriptors. J Mol Graph Model 18(4–5):464–477. https://doi.org/10.1016/s1093-3263(00)00068-1
136. Nantasenamat C, Isarankura-Na-Ayudhya C, Naenna T, Prachayasittikul V (2009) A practical overview of quantitative structure–activity relationship. EXCLI J 8:74–88. https://doi.org/10.17877/DE290R-690
137. Nantasenamat C, Isarankura-Na-Ayudhya C, Prachayasittikul V (2010) Advances in computational methods to predict the biological activity of compounds. Expert Opin Drug Discov 5(7):633–654. https://doi.org/10.1517/17460441.2010.492827
138. Randić M (2001) Novel shape descriptors for molecular graphs. J Chem Inform Comput Sci 41(3):607–613. https://doi.org/10.1021/ci0001031
139. Senese CL, Duca J, Pan D, Hopfinger AJ, Tseng YJ (2004) 4D-fingerprints, universal QSAR and QSPR descriptors. J Chem Inform Comput Sci 44(5):1526–1539. https://doi.org/10.1021/ci049898s
140. Shoombuatong W, Prathipati P, Owasirikul W, Worachartcheewan A, Simeon S, Anuwongcharoen N, Wikberg JES, Nantasenamat C (2017) Towards the revival of interpretable QSAR models. In: Roy K (ed) Advances in QSAR modeling challenges and advances in computational chemistry and physics, vol 24. Springer, Cham, pp 3–55. https://doi.org/10.1007/978-3-319-56850-8_1
141. Hawkins DM, Basak SC, Shi X (2001) QSAR with few compounds and many features. J Chem Inform Comput Sci 41(3):663–670. https://doi.org/10.1021/ci0001177
142. Rücker C, Rücker G, Meringer M (2007) y-randomization and its variants in QSPR/QSAR. J Chem Inform Model 47(6):2345–2357. https://doi.org/10.1021/ci700157b
143. Weaver S, Gleeson MP (2008) The importance of the domain of applicability in QSAR modeling. J Mol Graph Model 26(8):1315–1326. https://doi.org/10.1016/j.jmgm.2008.01.002
144. Gleeson MP, Modi S, Bender A, Robinson RLM, Kirchmair J, Promkatkaew M, Hannongbua S, Glen RC (2012) The challenges involved in modeling toxicity data in silico: a review. Curr Pharm Des 18(9):1266–1291. https://doi.org/10.2174/138161212799436359
145. Konovalov DA, Llewellyn LE, Vander Heyden Y, Coomans D (2008) Robust cross-validation of linear regression QSAR models. J Chem Inform Model 48(10):2081–2094. https://doi.org/10.1021/ci800209k
146. Eklund M, Norinder U, Boyer S, Carlsson L (2012) Application of conformal prediction in QSAR. IFIP Adv Inform Commun Technol 382:166–175. https://doi.org/10.1007/978-3-642-33412-2_17
147. Bosc N, Atkinson F, Felix E, Gaulton A, Hersey A, Leach AR (2019) Large scale comparison of QSAR and conformal prediction methods and their applications in drug discovery. J Cheminform 11(1):4. https://doi.org/10.1186/s13321-018-0325-4
148. Gleeson MP, Montanari D (2012) Strategies for the generation, validation and application of *in silico* ADMET models in lead generation and optimization. Exp Opin Drug Metab Toxicol 8(11):1435–1446. https://doi.org/10.1517/17425255.2012.711317
149. Topliss JG, Edwards RP (1979) Chance factors in studies of quantitative structure–activity relationships. J Med Chem 22(10):1238–1244. https://doi.org/10.1021/jm00196a017
150. Lombardo F, Gifford E, Shalaeva MY (2003) In silico ADME prediction: data, models, facts and myths. Mini Rev Med Chem 3(8):861–875. https://doi.org/10.2174/1389557033487629
151. Wood DJ, Buttar D, Cumming JG, Davis AM, Norinder U, Rodgers SL (2011) Automated QSAR with a hierarchy of global and local models.

Mol Inform 30(11–12):960–972. https://doi.org/10.1002/minf.201100107
152. Tetko IV, Bruneau P, Mewes H-W, Rohrer DC, Poda GI (2006) Can we estimate the accuracy of adme-tox predictions? Drug Disc Today 11(15–16):700–707. https://doi.org/10.1016/j.drudis.2006.06.013
153. 37th Joint Meeting of the Chemicals Committee (2004) OECD principles for the validation, for regulatory purposes, of (quantitative) structure–activity relationship models. https://www.oecd.org/chemicalsafety/risk-assessment/37849783.pdf. Accessed 9 Jan 2019
154. Judson PN, Barber C, Canipa SJ, Poignant G, Williams R (2015) Establishing good computer modelling practice (gcmp) in the prediction of chemical toxicity. Mol Inform 34(5):276–283. https://doi.org/10.1002/minf.201400137
155. Tropsha A (2010) Best practices for QSAR model development, validation, and exploitation. Mol Inform 29(6–7):476–488. https://doi.org/10.1002/minf.201000061
156. Patel M, Chilton ML, Sartini A, Gibson L, Barber C, Covey-Crump L, Przybylak KR, Cronin MTD, Madden JC (2018) Assessment and reproducibility of quantitative structure–activity relationship models by the nonexpert. J Chem Inform Model 58(3):673–682. https://doi.org/10.1021/acs.jcim.7b00523
157. Arora PK, Patil VM, Gupta SP (2010) A QSAR study on some series of anti-hepatitis B virus (HBV) agents. Bioinformation 4(9):417–420. https://doi.org/10.6026/97320630004417
158. Kurdekar V, Jadhav HR (2015) A new open source data analysis python script for QSAR study and its validation. Med Chem Res 24(4):1617–1625. https://doi.org/10.1007/s00044-014-1240-5
159. Research Collaboratory for Structural Bioinformatics (2019) The Protein Data Bank (PDB). http://www.rcsb.org/pdb/. Accessed 9 Jan 2019
160. Fiser A, Sali A (2003) Modeller: generation and refinement of homology-based protein structure models. Methods Enzymol 374:461–491. https://doi.org/10.1016/S0076-6879(03)74020-8
161. Ewing TJ, Makino S, Skillman AG, Kuntz ID (2001) Dock 4.0: search strategies for automated molecular docking of flexible molecule databases. J Comput Aided Mol Des 15(5):411–428. https://doi.org/10.1023/a:1011115820450
162. Goodsell DS, Olson AJ (1990) Automated docking of substrates to proteins by simulated annealing. Proteins 8(3):195–202. https://doi.org/10.1002/prot.340080302
163. Warren GL, Andrews CW, Capelli AM, Clarke B, LaLonde J, Lambert MH, Lindvall M, Nevins N, Semus SF, Senger S, Tedesco G, Wall ID, Woolven JM, Peishoff CE, Head MS (2006) A critical assessment of docking programs and scoring functions. J Med Chem 49(20):5912–5931. https://doi.org/10.1021/jm050362n
164. Kubinyi H (1997) QSAR and 3D QSAR in drug design Part 2: applications and problems. Drug Discov Today 2:538–546. https://doi.org/10.1016/S1359-6446(97)01084-2
165. Kubinyi H (1997) QSAR and 3D QSAR in drug design Part 1: methodology. Drug Discov Today 2(11):457–467. https://doi.org/10.1016/S1359-6446(97)01079-9
166. Cramer RD, Wendt B (2007) Pushing the boundaries of 3D-QSAR. J Comput Aided Mol Des 21(1–3):23–32. https://doi.org/10.1007/s10822-006-9100-0
167. Leach AR (2001) Molecular modelling: principles and applications, 2nd edn. Pearson Education, Harlow
168. Menikarachchi LC, Gascón JA (2010) QM/MM approaches in medicinal chemistry research. Curr Top Med Chem 10(1):46–54. https://doi.org/10.2174/156802610790232297
169. Mulholland AJ (2007) Chemical accuracy in QM/MM calculations on enzyme-catalysed reactions. Chem Cent J 1:19. https://doi.org/10.1186/1752-153X-1-19
170. Senn HM, Thiel W (2007) QM/MM studies of enzymes. Curr Opin Chem Biol 11(2):182–187. https://doi.org/10.1016/j.cbpa.2007.01.684
171. Senn HM, Thiel W (2009) QM/MM methods for biomolecular systems. Angewandte Chemie 48(7):1198–1229. https://doi.org/10.1002/anie.200802019
172. Walker RC, Crowley MF, Case DA (2008) The implementation of a fast and accurate QM/MM potential method in amber. J Comput Chem 29(7):1019–1031. https://doi.org/10.1002/jcc.20857
173. Butcher EC, Berg EL, Kunkel EJ (2004) Systems biology in drug discovery. Nat Biotechnol 22(10):1253–1259. https://doi.org/10.1038/nbt1017
174. Pujol A, Mosca R, Farres J, Aloy P (2010) Unveiling the role of network and systems biology in drug discovery. Trends Pharmacol Sci 31(3):115–123. https://doi.org/10.1016/j.tips.2009.11.006
175. Keiser MJ, Setola V, Irwin JJ, Laggner C, Abbas AI, Hufeisen SJ, Jensen NH, Kuijer MB, Matos RC, Tran TB, Whaley R, Glennon RA, Hert J, Thomas KL, Edwards DD, Shoichet BK, Roth BL (2009) Predicting new molecular targets for known drugs. Nature 462(7270):175–181. https://doi.org/10.1038/nature08506
176. Ye H, Wei J, Tang K, Feuers R, Hong H (2016) Drug repositioning through network pharmacology. Curr Top Med Chem 16(30):3646–3656. https://doi.org/10.2174/1568026616666160530181328
177. Keiser MJ, Roth BL, Armbruster BN, Ernsberger P, Irwin JJ, Shoichet BK (2007) Relating protein pharmacology by ligand chemistry. Nat Biotechnol 25(2):197–206. https://doi.org/10.1038/nbt1284
178. Wu W, Zhang R, Salahub DR (2009) Nelfinavir: a magic bullet to annihilate cancer cells? Cancer Biol Ther 8(3):233–235. https://doi.org/10.4161/cbt.8.3.7789
179. Dakshanamurthy S, Issa NT, Assefnia S, Seshasayee A, Peters OJ, Madhavan S, Uren A, Brown ML, Byers SW (2012) Predicting new indications for approved drugs using a proteochemometric method. J Med Chem 55(15):6832–6848. https://doi.org/10.1021/jm300576q
180. Schaduangrat N, Anuwongcharoen N, Phanus-umporn C, Sriwanichpoom N, Wikberg JES, Nantasenamat C (2019) Chapter 10—Proteochemometric modeling for drug repositioning. In: Roy K (ed) In Silico Drug Design. Academic Press, London, pp 281–302. https://doi.org/10.1016/B978-0-12-816125-8.00010-9
181. Waltemath D, Wolkenhauer O (2016) How modeling standards, software, and initiatives support reproducibility in systems biology and systems medicine. IEEE Trans Biomed Eng 63(10):1999–2006. https://doi.org/10.1109/TBME.2016.2555481
182. Medley JK, Goldberg AP, Karr JR (2016) Guidelines for reproducibly building and simulating systems biology models. IEEE Trans Biomed Eng 63(10):2015–2020. https://doi.org/10.1109/TBME.2016.2591960
183. Waltemath D, Henkel R, Winter F, Wolkenhauer O (2013) Reproducibility of model-based results in systems biology. In: Prokop A, Csukás B (eds) Syst Biol. Springer, Dordrecht, pp 301–320. https://doi.org/10.1007/978-94-007-6803-1_10
184. Le Novère N, Bornstein B, Broicher A, Courtot M, Donizelli M, Dharuri H, Li L, Sauro H, Schilstra M, Shapiro B, Snoep JL, Hucka M (2006) BioModels database: a free, centralized database of curated, published, quantitative kinetic models of biochemical and cellular systems. Nucleic Acids Res 34:689–691. https://doi.org/10.1093/nar/gkj092
185. Kirouac DC, Cicali B, Schmidt S (2019) Reproducibility of quantitative systems pharmacology models: current challenges and future opportunities. CPT Pharmacometrics Syst Pharmacol 8(4):205–210. https://doi.org/10.1002/psp4.12390
186. Watanabe L, Barhak J, Myers C (2019) Toward reproducible disease models using the systems biology markup language. Simulation 95(10):895–930. https://doi.org/10.1177/0037549718793214
187. Hucka M, Finney A, Sauro HM, Bolouri H, Doyle JC, Kitano H, Arkin AP, Bornstein BJ, Bray D, Cornish-Bowden A, Cuellar AA, Dronov S, Gilles ED, Ginkel M, Gor V, Goryanin II, Hedley WJ, Hodgman TC, Hofmeyr JH, Hunter PJ, Juty NS, Kasberger JL, Kremling A, Kummer U, Le Novere N, Loew LM, Lucio D, Mendes P, Minch E, Mjolsness ED, Nakayama Y, Nelson MR, Nielsen PF, Sakurada T, Schaff JC, Shapiro BE, Shimizu TS, Spence HD, Stelling J, Takahashi K, Tomita M, Wagner J, Wang J (2003) The systems biology markup language (SBML): a medium for representation and exchange of biochemical network models. Bioinformatics 19(4):524–531. https://doi.org/10.1093/bioinformatics/btg015
188. Swat MJ, Moodie S, Wimalaratne SM, Kristensen NR, Lavielle M, Mari A, Magni P, Smith MK, Bizzotto R, Pasotti L, Mezzalana E, Comets E, Sarr C, Terranova N, Blaudez E, Chan P, Chard J, Chatel K, Chenel M, Edwards D, Franklin C, Giorgino T, Glont M, Girard P, Grenon P, Harling K, Hooker AC, Kaye R, Keizer R, Kloft C, Kok JN, Kokash N, Laibe C, Laveille C, Lestini G, Mentre F, Munafo A, Nordgren R, Nyberg HB, Parra-Guillen ZP, Plan E, Ribba B, Smith G, Troconiz IF, Yvon F, Milligan PA, Harnisch L, Karlsson M, Hermjakob H, Le Novere N (2015) Pharmacometrics Markup Language (PharmML): opening new perspectives for model exchange in drug development. CPT Pharmacometrics Syst Pharmacol 4(6):316–319. https://doi.org/10.1002/psp4.57

189. Barhak J (2019) MIST: Micro-simulation tool to support disease modeling. https://github.com/scipy-conference/scipy2013_talks/tree/master/talks/jacob_barhak. Accessed 1 Nov 2019
190. Hedley WJ, Nelson MR, Bullivant DP, Nielsen PF (2001) A short introduction to cellML. Philos Trans R Soc A 359(1783):1073–1089. https://doi.org/10.1098/rsta.2001.0817
191. Medley JK, Choi K, Konig M, Smith L, Gu S, Hellerstein J, Sealfon SC, Sauro HM (2018) Tellurium notebooks—an environment for reproducible dynamical modeling in systems biology. PLoS Comput Biol 14(6):1006220. https://doi.org/10.1371/journal.pcbi.1006220
192. Choi K, Medley JK, Konig M, Stocking K, Smith L, Gu S, Sauro HM (2018) Tellurium: an extensible python-based modeling environment for systems and synthetic biology. BioSystems 171:74–79. https://doi.org/10.1016/j.biosystems.2018.07.006
193. Kolpakov F, Akberdin I, Kashapov T, Kiselev L, Kolmykov S, Kondrakhin Y, Kutumova E, Mandrik N, Pintus S, Ryabova A, Sharipov R, Yevshin I, Kel A (2019) BioUML: an integrated environment for systems biology and collaborative analysis of biomedical data. Nucleic Acids Res 47(W1):225–233. https://doi.org/10.1093/nar/gkz440
194. Drawert B, Trogdon M, Toor S, Petzold L, Hellander A (2016) MOLNs: A cloud platform for interactive, reproducible, and scalable spatial stochastic computational experiments in systems biology using PyURDME. SIAM J Sci Comput 38(3):179–202. https://doi.org/10.1137/15M1014784
195. Schadt EE, Linderman MD, Sorenson J, Lee L, Nolan GP (2010) Computational solutions to large-scale data management and analysis. Nat Rev Genet 11(9):647–657. https://doi.org/10.1038/nrg2857
196. Noble WS (2009) A quick guide to organizing computational biology projects. PLoS Comput Biol 5(7):1000424. https://doi.org/10.1371/journal.pcbi.1000424
197. Hassan M, Brown RD, VarmaO'brien S, Rogers D (2006) Cheminformatics analysis and learning in a data pipelining environment. Mol Divers 10(3):283–299. https://doi.org/10.1007/s11030-006-9041-5
198. Berthold MR, Cebron N, Dill F, Gabriel TR, Kötter T, Meinl T, Ohl P, Thiel K, Wiswedel B (2009) KNIME—the Konstanz information miner. ACM SIGKDD Explor Newslett 11(1):26. https://doi.org/10.1145/1656274.1656280
199. Cox R, Green DVS, Luscombe CN, Malcolm N, Pickett SD (2013) QSAR workbench: automating QSAR modeling to drive compound design. J Comput Aided Mol Des 27(4):321–336. https://doi.org/10.1007/s10822-013-9648-4
200. Steinmetz FP, Mellor CL, Meinl T, Cronin MTD (2015) Screening chemicals for receptor-mediated toxicological and pharmacological endpoints: using public data to build screening tools within a KNIME workflow. Mol Inform 34(2–3):171–178. https://doi.org/10.1002/minf.201400188
201. Nicola G, Berthold MR, Hedrick MP, Gilson MK (2015) Connecting proteins with drug-like compounds: open source drug discovery workflows with BindingDB and KNIME. Database. https://doi.org/10.1093/database/bav087
202. Mazanetz MP, Marmon RJ, Reisser CBT, Morao I (2012) Drug discovery applications for knime: an open source data mining platform. Curr Top Med Chem 12(18):1965–1979. https://doi.org/10.2174/156802612804910331
203. Kuhn T, Willighagen EL, Zielesny A, Steinbeck C (2010) Cdk-taverna: an open workflow environment for cheminformatics. BMC Bioinform 11:159. https://doi.org/10.1186/1471-2105-11-159
204. Steinbeck C, Han Y, Kuhn S, Horlacher O, Luttmann E, Willighagen E (2003) The Chemistry Development Kit (CDK): an open-source Java Library for Chemo- and Bioinformatics. J Chem Inform Comput Sci 43(2):493–500. https://doi.org/10.1021/ci025584y
205. Willighagen EL, Mayfield JW, Alvarsson J, Berg A, Carlsson L, Jeliazkova N, Kuhn S, Pluskal T, Rojas-Chertó M, Spjuth O, Torrance G, Evelo CT, Guha R, Steinbeck C (2017) The Chemistry Development Kit (CDK) v2.0: atom typing, depiction, molecular formulas, and substructure searching. J Cheminform 9:33. https://doi.org/10.1186/s13321-017-0220-4
206. Lucas X, Grüning BA, Günther S (2014) ChemicalToolBoX and its application on the study of the drug like and purchasable space. J Cheminform 6(Suppl 1):51. https://doi.org/10.1186/1758-2946-6-S1-P51
207. Di Tommaso P, Chatzou M, Floden EW, Barja PP, Palumbo E, Notredame C (2017) Nextflow enables reproducible computational workflows. Nat Biotechnol 35(4):316–319. https://doi.org/10.1038/nbt.3820
208. Köster J, Rahmann S (2012) Snakemake-a scalable bioinformatics workflow engine. Bioinformatics 28(19):2520–2522. https://doi.org/10.1093/bioinformatics/bts480
209. Goodstadt L (2010) Ruffus: a lightweight python library for computational pipelines. Bioinformatics 26(21):2778–2779. https://doi.org/10.1093/bioinformatics/btq524
210. Sadedin SP, Pope B, Oshlack A (2012) Bpipe: a tool for running and managing bioinformatics pipelines. Bioinformatics 28(11):1525–1526. https://doi.org/10.1093/bioinformatics/bts167
211. Brandt J, Reisig W, Leser ULF (2017) Computation semantics of the functional scientific workflow language cuneiform. J Funct Program. https://doi.org/10.1017/S0956796817000119
212. Bernhardsson E, Freider E, Rouhani A (2012) Luigi GitHub repository. https://github.com/spotify/luigi
213. Wilson G, Aruliah DA, Brown CT, Chue Hong NP, Davis M, Guy RT, Haddock SH, Huff KD, Mitchell IM, Plumbley MD, Waugh B, White EP, Wilson P (2014) Best practices for scientific computing. PLoS Biol 12(1):1001745. https://doi.org/10.1371/journal.pbio.1001745
214. Taschuk M, Wilson G (2017) Ten simple rules for making research software more robust. PLoS Comput Biol 13(4):1005412. https://doi.org/10.1371/journal.pcbi.1005412
215. Nowotka MM, Gaulton A, Mendez D, Bento AP, Hersey A, Leach A (2017) Using ChEMBL web services for building applications and data processing workflows relevant to drug discovery. Exp Opin Drug Discov 12(8):757–767. https://doi.org/10.1080/17460441.2017.1339032
216. Alvarsson J, Lampa S, Schaal W, Andersson C, Wikberg JES, Spjuth O (2016) Large-scale ligand-based predictive modelling using support vector machines. J Cheminform 8:39. https://doi.org/10.1186/s13321-016-0151-5
217. Lampa S, Alvarsson J, Spjuth O (2016) Towards agile large-scale predictive modelling in drug discovery with flow-based programming design principles. J Cheminform 8:67. https://doi.org/10.1186/s13321-016-0179-6
218. Yoo AB, Jette MA, Grondona M (2003) SLURM: simple linux utility for resource management. In: Feitelson D, Rudolph L, Schwiegelshohn U (eds) Job scheduling strategies for parallel processing. Lecture notes in computer science, vol 2862. Springer, Berlin, pp 44–60
219. Amstutz P, Crusoe MR, Tijanić N, Chapman B, Chilton J, Heuer M, Kartashov A, Leehr D, Ménager H, Nedeljkovich M, Scales M, Soiland-Reyes S, Stojanovic L (2019) Common Workflow Language, v1.0. https://doi.org/10.6084/m9.figshare.3115156.v2. Accessed 9 Jan 2019
220. Chapman B, Gentry J, Lin M, Magee P, O'Connor B, Prabhakaran A, Van der Auwera G (2019) OpenWDL. http://www.openwdl.org/. Accessed 9 Jan 2019
221. Davie P (2010) Cloud computing: a drug discovery game changer? Innov Pharm Technol 33:34–36
222. Dudley JT, Butte AJ (2010) *In silico* research in the era of cloud computing. Nat Biotechnol 28(11):1181–1185. https://doi.org/10.1038/nbt1110-1181
223. Garg V, Arora S, Gupta C (2011) Cloud computing approaches to accelerate drug discovery value chain. Comb Chem High Throughput Screen 14(10):861–871. https://doi.org/10.2174/138620711797537085
224. Moghadam BT, Alvarsson J, Holm M, Eklund M, Carlsson L, Spjuth O (2015) Scaling predictive modeling in drug development with cloud computing. J Chem Inform Model 55(1):19–25. https://doi.org/10.1021/ci500580y
225. Hurley DG, Budden DM, Crampin EJ (2015) Virtual reference environments: a simple way to make research reproducible. Brief Bioinform 16(5):901–903. https://doi.org/10.1093/bib/bbu043
226. Piccolo SR, Frampton MB (2016) Tools and techniques for computational reproducibility. GigaScience 5(1):30. https://doi.org/10.1186/s13742-016-0135-4
227. Jaghoori MM, Bleijlevens B, Olabarriaga SD (2016) 1001 ways to run AutoDock Vina for virtual screening. J Comput Aided Mol Des 30(3):237–249. https://doi.org/10.1007/s10822-016-9900-9
228. McGuire R, Verhoeven S, Vass M, Vriend G, de Esch IJ, Lusher SJ, Leurs R, Ridder L, Kooistra AJ, Ritschel T, de Graaf C (2017) 3D-e-Chem-VM: structural cheminformatics research infrastructure in a freely available virtual machine. J Chem Inf Model 57(2):115–121. https://doi.org/10.1021/acs.jcim.6b00686

229. Alvim-Gaston M, Grese T, Mahoui A, Palkowitz AD, Pineiro-Nunez M, Watson I (2014) Open Innovation Drug Discovery (OIDD): a potential path to novel therapeutic chemical space. Curr Top Med Chem 14(3):294–303. https://doi.org/10.2174/1568026613666131127125858
230. Ochoa R, Davies M, Papadatos G, Atkinson F, Overington JP (2014) myChEMBL: a virtual machine implementation of open data and cheminformatics tools. Bioinformatics 30(2):298–300. https://doi.org/10.1093/bioinformatics/btt666
231. Ellingson SR, Baudry J (2011) High-throughput virtual molecular docking: Hadoop implementation of AutoDock4 on a private cloud. In: Proceedings of the second international workshop on emerging computational methods for the life sciences - ECMLS'11. ACM Press, New York, pp 33–38. https://doi.org/10.1145/1996023.1996028
232. Capuccini M, Ahmed L, Schaal W, Laure E, Spjuth O (2017) Large-scale virtual screening on public cloud resources with apache spark. J Cheminform 9:15. https://doi.org/10.1186/s13321-017-0204-4
233. Georgieva P, Lapins M, Spjuth O, Wikberg J (2019) Pharmaceutical bioinformatics: A free internet course for international and Swedish students offered by the University of Uppsala. http://www.pharmbio.org/. Accessed 1 Nov 2019
234. Dahlö M, Haziza F, Kallio A, Korpelainen E, Bongcam-Rudloff E, Spjuth O (2015) BioImg.org: a catalog of virtual machine images for the life sciences. Bioinform Biol Insights 9:125–128. https://doi.org/10.4137/BBI.S28636
235. Cito J, Gall HC (2016) Using docker containers to improve reproducibility in software engineering research. In: Proceedings of the 38th international conference on software engineering companion—ICSE '16. ACM Press, New York, pp 906–907
236. Silver A (2017) Software simplified. Nature 546(7656):173–174. https://doi.org/10.1038/546173a
237. Kurtzer GM, Sochat V, Bauer MW (2017) Singularity: Scientific containers for mobility of compute. PLoS ONE 12(5):0177459. https://doi.org/10.1371/journal.pone.0177459
238. Gomes J, Campos I, Bagnaschi E, David M, Alves L, Martins J, Pina J, Lopez-Garcia A, Orviz P (2017) Enabling rootless linux containers in multi-user environments: the *udocker* tool. Comput Phys Commun 232:84–97. https://doi.org/10.1016/j.cpc.2018.05.021
239. Warr WA (2012) Scientific workflow systems: pipeline pilot and knime. J Comput Aided Mol Des 26(7):801–804. https://doi.org/10.1007/s10822-012-9577-7
240. Suhartanto H, Pasaribu AP, Siddiq MF, Fadhila MI, Hilman MH, Yanuar A (2017) A preliminary study on shifting from virtual machine to docker container for insilico drug discovery in the cloud. Int J Technol 8(4):611. https://doi.org/10.14716/ijtech.v8i4.9478
241. Fong J (2019) How GlaxoSmithKline is Accelerating Science with Docker Enterprise Edition. https://blog.docker.com/2017/10/how-gsk-is-accelerating-science-with-dockeree/. Accessed 9 Jan 2019
242. Altae-Tran H, Ramsundar B, Pappu AS, Pande V (2017) Low data drug discovery with one-shot learning. ACS Cent Sci 3(4):283–293. https://doi.org/10.1021/acscentsci.6b00367
243. OpenRiskNet (2019) Open e-infrastructure to support data sharing, knowledge integration and in silico analysis and modelling in predictive toxicology and risk assessment. http://www.openrisknet.org/. Accessed 9 Jan 2019
244. Belmann P, Dröge J, Bremges A, McHardy AC, Sczyrba A, Barton MD (2015) Bioboxes: standardised containers for interchangeable bioinformatics software. GigaScience 4:47. https://doi.org/10.1186/s13742-015-0087-0
245. Li W, Kanso A (2015) Comparing containers versus virtual machines for achieving high availability. In: 2015 IEEE international conference on cloud engineering. IEEE, New Jersey, pp 353–358. https://doi.org/10.1109/IC2E.2015.79
246. Spjuth O, Willighagen EL, Guha R, Eklund M, Wikberg JE (2010) Towards interoperable and reproducible QSAR analyses: exchange of datasets. J Cheminform 2(1):5. https://doi.org/10.1186/1758-2946-2-5
247. Ruusmann V, Sild S, Maran U (2014) QSAR databank—an approach for the digital organization and archiving of QSAR model information. J Cheminform 6:25. https://doi.org/10.1186/1758-2946-6-25
248. Ruusmann V, Sild S, Maran U (2015) QSAR databank repository: open and linked qualitative and quantitative structure-activity relationship models. J Cheminform 7(1):32. https://doi.org/10.1186/s13321-015-0082-6
249. Joint Research Centre, The European's Commission's science and knowledge service (2019) (Q)SAR Model Reporting Format Database. https://qsardb.jrc.ec.europa.eu/qmrf/. Accessed 1 Nov 2019
250. Hastings J, Jeliazkova N, Owen G, Tsiliki G, Munteanu CR, Steinbeck C, Willighagen E (2015) eNanoMapper: harnessing ontologies to enable data integration for nanomaterial risk assessment. J Biomed Demant 6(1):10
251. Guazzelli A, Zeller M, Lin W-C, Williams G et al (2009) PMML: an open standard for sharing models. R J 1(1):60–65
252. Center for Computational Science Research, Inc. (2019) Data Mining Group. http://dmg.org/. Accessed 1 Nov 2019
253. Fillbrunn A (2019) PMML integration in KNIME. https://www.knime.com/blog/pmml-integration-in-knime/. Accessed 1 Nov 2019
254. ONNX Project Contributors (2019) Open Neural Network Exchange Format: The open ecosystem for interchangeable AI models. https://onnx.ai/. Accessed 1 Nov 2019
255. Stålring JC, Carlsson LA, Almeida P, Boyer S (2011) AZOrange—high performance open source machine learning for QSAR modeling in a graphical programming environment. J Cheminform 3:28. https://doi.org/10.1186/1758-2946-3-28
256. Dixon SL, Duan J, Smith E, Von Bargen CD, Sherman W, Repasky MP (2016) AutoQSAR: an automated machine learning tool for best-practice quantitative structure-activity relationship modeling. Fut Med Chem 8(15):1825–1839. https://doi.org/10.4155/fmc-2016-0093
257. Nantasenamat C, Worachartcheewan A, Jamsak S, Preeyanon L, Shoombuatong W, Simeon S, Mandi P, Isarankura-Na-Ayudhya C, Prachayasittikul V (2015) AutoWeka: toward an automated data mining software for QSAR and QSPR studies. Methods Mol Biol 1260:119–147. https://doi.org/10.1007/978-1-4939-2239-0_8
258. Hall M, Frank E, Holmes G, Pfahringer B, Reutemann P, Witten IH (2009) The weka data mining software. ACM SIGKDD Explor Newslett 11(1):10. https://doi.org/10.1145/1656274.1656278
259. Kausar S, Falcao AO (2018) An automated framework for QSAR model building. J Cheminform 10(1):1. https://doi.org/10.1186/s13321-017-0256-5
260. Dong J, Yao Z-J, Zhu M-F, Wang N-N, Lu B, Chen AF, Lu A-P, Miao H, Zeng W-B, Cao D-S (2017) ChemSAR: an online pipelining platform for molecular SAR modeling. J Cheminform 9(1):27. https://doi.org/10.1186/s13321-017-0215-1
261. Tsiliki G, Munteanu CR, Seoane JA, Fernandez-Lozano C, Sarimveis H, Willighagen EL (2015) Rregrs: an r package for computer-aided model selection with multiple regression models. J Cheminform 7:46. https://doi.org/10.1186/s13321-015-0094-2
262. Murrell DS, Cortes-Ciriano I, van Westen GJP, Stott IP, Bender A, Malliavin TE, Glen RC (2015) Chemically aware model builder (camb): an r package for property and bioactivity modelling of small molecules. J Cheminform 7:45. https://doi.org/10.1186/s13321-015-0086-2
263. Shamsara J (2017) Ezqsar: an R package for developing QSAR models directly from structures. Open Med Chem J 11:212–221. https://doi.org/10.2174/1874104501711010212
264. Nantasenamat C (2020) Best practices for constructing reproducible QSAR models. In: Roy K (ed) Ecotoxicological QSARs. Humana Press, New Jersey
265. Rule A, Birmingham A, Zuniga C, Altintas I, Huang S-C, Knight R, Moshiri N, Nguyen MH, Rosenthal SB, Pérez F, Rose PW (2019) Ten simple rules for writing and sharing computational analyses in jupyter notebooks. PLoS Comput Biol 15(7):1007007
266. Landrum G (2019) RDKit tutorials. Available online: https://github.com/greglandrum/. Accessed 1 Nov 2019
267. RDKit (2019) RDKit: Open-Source Cheminformatics Software. https://www.rdkit.org/. Accessed 1 Nov 2019
268. RDKit GitHub (2019) RDKit. https://github.com/rdkit/rdkit-tutorials/. Accessed 1 Nov 2019
269. OpenEye Scientific Software, Inc (2019) OpenEye Python Cookbook. https://docs.eyesopen.com/toolkits/cookbook/python/. Accessed 1 Nov 2019
270. Informatics Matters Ltd (2019) Squonk Computational Notebook. https://squonk.it/. Accessed 1 Nov 2019

271. CDK (2019) Chemistry Development Kit: Open Source modular Java libraries for Cheminformatics. https://cdk.github.io/. Accessed 1 Nov 2019
272. Jansen JM, Cornell W, Tseng YJ, Amaro RE (2012) Teach-Discover-Treat (TDT): collaborative computational drug discovery for neglected diseases. J Mol Graph Model 38:360–362. https://doi.org/10.1016/j.jmgm.2012.07.007
273. Riniker S, Landrum GA, Montanari F, Villalba SD, Maier J, Jansen JM, Walters WP, Shelat AA (2017) Virtual-screening workflow tutorials and prospective results from the Teach-Discover-Treat competition 2014 against malaria. F1000 Res 6:1136. https://doi.org/10.12688/f1000research.11905.2
274. Riniker S, Landrum GA, Montanari F, Villalba SD, Maier J, Jansen, JM, Walters WP, Shelat AA (2019) Tutorial for the Teach-Discover-Treat (TDT) competition 2014-Challenge 1: anti-malaria hit finding using classifier-fusion boosted predictive models. https://github.com/sriniker/TDT-tutorial-2014/. Accessed 1 Nov 2019
275. Sydow D, Morger A, Driller M, Volkamer A (2019) TeachOpenCADD: a teaching platform for computer-aided drug design using open source packages and data. J Cheminform 11:29. https://doi.org/10.1186/s13321-019-0351-x
276. Kluyver T, Ragan-Kelley B, Pérez F, Granger B, Bussonnier M, Frederic J, Kelley K, Hamrick J, Grout J, Corlay S, Ivanov P, Avila D, Abdalla S, Willing C (2016) development team, J.: Jupyter notebooks - a publishing format for reproducible computational workflows. In: Loizides F, Scmidt B (eds) Positioning and power in Academic Publishing: players, agents and agendas. IOS Press, Amsterdam, pp 87–90. https://eprints.soton.ac.uk/403913/
277. Grünberg R, Nilges M, Leckner J (2007) Biskit-a software platform for structural bioinformatics. Bioinformatics 23(6):769–770. https://doi.org/10.1093/bioinformatics/btl655
278. Daniluk P, Wilczyński B, Lesyng B (2015) WeBIAS: a web server for publishing bioinformatics applications. BMC Res Notes 8:628. https://doi.org/10.1186/s13104-015-1622-x
279. Osz Á, Pongor LS, Szirmai D, Gyorffy B (2017) A snapshot of 3649 web-based services published between 1994 and 2017 shows a decrease in availability after 2 years. Brief Bioinform. https://doi.org/10.1093/bib/bbx159
280. RStudio Inc. (2018) Shiny. https://shiny.rstudio.com/
281. Plotly (2019) Dash. https://plot.ly/products/dash/. Accessed 9 Jan 2019
282. Plotly (2019) Plotly: Modern analytic apps for the enterprise. https://plot.ly/. Accessed 9 Jan 2019
283. Nantasenamat C (2019) Conceptual map of computational drug discovery [CC-BY]. https://doi.org/10.6084/m9.figshare.5979400
284. Synergy Research Group (2019) The leading cloud providers continue to run away with the market. https://www.srgresearch.com/articles/leading-cloud-providers-continue-run-away-market/. Accessed 9 Jan 2019
285. Dong J, Yao Z-J, Wen M, Zhu M-F, Wang N-N, Miao H-Y, Lu A-P, Zeng W-B, Cao D-S (2016) Biotriangle: a web-accessible platform for generating various molecular representations for chemicals, proteins, dnas/rnas and their interactions. J Cheminform 8:34. https://doi.org/10.1186/s13321-016-0146-2
286. Dong J, Cao D-S, Miao H-Y, Liu S, Deng B-C, Yun Y-H, Wang N-N, Lu A-P, Zeng W-B, Chen AF (2015) Chemdes: an integrated web-based platform for molecular descriptor and fingerprint computation. J Cheminform 7:60. https://doi.org/10.1186/s13321-015-0109-z
287. Walker T, Grulke CM, Pozefsky D, Tropsha A (2010) Chembench: a cheminformatics workbench. Bioinformatics 26(23):3000–3001. https://doi.org/10.1093/bioinformatics/btq556
288. Sushko I, Novotarskyi S, Körner R, Pandey AK, Rupp M, Teetz W, Brandmaier S, Abdelaziz A, Prokopenko VV, Tanchuk VY et al (2011) Online chemical modeling environment (OCHEM): web platform for data storage, model development and publishing of chemical information. J Comput Aided Mol Des 25(6):533–554. https://doi.org/10.1007/s10822-011-9440-2
289. González-Medina M, Medina-Franco JL (2017) Platform for unified molecular analysis: Puma. J Chem Inform Model 57(8):1735–1740. https://doi.org/10.1021/acs.jcim.7b00253
290. van Zundert GCP, Rodrigues JPGLM, Trellet M, Schmitz C, Kastritis PL, Karaca E, Melquiond ASJ, van Dijk M, de Vries SJ, Bonvin AMJJ (2016) The haddock2.2 web server: user-friendly integrative modeling of biomolecular complexes. J Mol Biol 428(4):720–725. https://doi.org/10.1016/j.jmb.2015.09.014
291. Camps J, Carrillo O, Emperador A, Orellana L, Hospital A, Rueda M, Cicin-Sain D, D'Abramo M, Gelpí JL, Orozco M (2009) FlexServ: an integrated tool for the analysis of protein flexibility. Bioinformatics 25(13):1709–1710. https://doi.org/10.1093/bioinformatics/btp304
292. Hospital A, Andrio P, Fenollosa C, Cicin-Sain D, Orozco M, Gelpí JL (2012) MDWeb and MDMoby: an integrated web-based platform for molecular dynamics simulations. Bioinformatics 28(9):1278–1279. https://doi.org/10.1093/bioinformatics/bts139
293. Stierand K, Maass PC, Rarey M (2006) Molecular complexes at a glance: automated generation of two-dimensional complex diagrams. Bioinformatics 22(14):1710–1716. https://doi.org/10.1093/bioinformatics/btl150
294. Biasini M, Bienert S, Waterhouse A, Arnold K, Studer G, Schmidt T, Kiefer F, Gallo Cassarino T, Bertoni M, Bordoli L, Torsten S (2014) Swiss-model: modelling protein tertiary and quaternary structure using evolutionary information. Nucleic Acids Res 42(Web Server issue):252–8. https://doi.org/10.1093/nar/gku340

Publisher's Note

Terms and Conditions

Springer Nature journal content, brought to you courtesy of Springer Nature Customer Service Center GmbH (“Springer Nature”).
Springer Nature supports a reasonable amount of sharing of research papers by authors, subscribers and authorised users (“Users”), for small-scale personal, non-commercial use provided that all copyright, trade and service marks and other proprietary notices are maintained. By accessing, sharing, receiving or otherwise using the Springer Nature journal content you agree to these terms of use (“Terms”). For these purposes, Springer Nature considers academic use (by researchers and students) to be non-commercial.
These Terms are supplementary and will apply in addition to any applicable website terms and conditions, a relevant site licence or a personal subscription. These Terms will prevail over any conflict or ambiguity with regards to the relevant terms, a site licence or a personal subscription (to the extent of the conflict or ambiguity only). For Creative Commons-licensed articles, the terms of the Creative Commons license used will apply.
We collect and use personal data to provide access to the Springer Nature journal content. We may also use these personal data internally within ResearchGate and Springer Nature and as agreed share it, in an anonymised way, for purposes of tracking, analysis and reporting. We will not otherwise disclose your personal data outside the ResearchGate or the Springer Nature group of companies unless we have your permission as detailed in the Privacy Policy.
While Users may use the Springer Nature journal content for small scale, personal non-commercial use, it is important to note that Users may not:

1. use such content for the purpose of providing other users with access on a regular or large scale basis or as a means to circumvent access control;
2. use such content where to do so would be considered a criminal or statutory offence in any jurisdiction, or gives rise to civil liability, or is otherwise unlawful;
3. falsely or misleadingly imply or suggest endorsement, approval , sponsorship, or association unless explicitly agreed to by Springer Nature in writing;
4. use bots or other automated methods to access the content or redirect messages
5. override any security feature or exclusionary protocol; or
6. share the content in order to create substitute for Springer Nature products or services or a systematic database of Springer Nature journal content.

In line with the restriction against commercial use, Springer Nature does not permit the creation of a product or service that creates revenue, royalties, rent or income from our content or its inclusion as part of a paid for service or for other commercial gain. Springer Nature journal content cannot be used for inter-library loans and librarians may not upload Springer Nature journal content on a large scale into their, or any other, institutional repository.
These terms of use are reviewed regularly and may be amended at any time. Springer Nature is not obligated to publish any information or content on this website and may remove it or features or functionality at our sole discretion, at any time with or without notice. Springer Nature may revoke this licence to you at any time and remove access to any copies of the Springer Nature journal content which have been saved.
To the fullest extent permitted by law, Springer Nature makes no warranties, representations or guarantees to Users, either express or implied with respect to the Springer nature journal content and all parties disclaim and waive any implied warranties or warranties imposed by law, including merchantability or fitness for any particular purpose.
Please note that these rights do not automatically extend to content, data or other material published by Springer Nature that may be licensed from third parties.
If you would like to use or distribute our Springer Nature journal content to a wider audience or on a regular basis or in any other manner not expressly permitted by these Terms, please contact Springer Nature at

onlineservice@springernature.com